Department of Informatics
School of Commercial Law and Economics
University of Gothenburg
Master Thesis, Autumn 2002

Designing a Meta-Architectural Support for Standard Systems Evaluation

A case study at OSS Sales, Ericsson Corporation.

Authors:
Martin Andersson
Darko Aleksic

Supervisor:
Maria Bergenstjerna

Abstract

The main purpose of this thesis is to contribute to an increased feeling of security and confidence in the process of making a sound choice of a standard system. We started our work with a theoretical study in order to learn about systems, in particular standard systems; methods, in particular methods in information systems development and methodological knowledge i.e. methods from a knowledge perspective. Further on, we assembled a tentative view of the unbalance between our need for knowledge and our accumulated knowledge. Relevant theories, i.e. methods relating to the field of standard systems evaluation, were analyzed and combined by method-fragments. This resulted in, by our means, a confident and secure meta-architecture for standard systems evaluation. We empirically tested the meta-architecture at OSS Sales that needed professional aid in making a sound choice of a standard system for request handling.

Keywords: Meta-architecture, Method, System development, Evaluation, Design, Standard systems.

Acknowledgements:

We would like to thank our supervisor, Maria Bergenstjerna, for her knowledge, support and encouragement in situations when we were in need of guidance. We also would like to thank B-O Hertz at OSS Sales, Ericsson AB in Gårda, Gothenburg for providing us with a case and letting us use Ericsson resources while writing our thesis.

Table of Content

1 Introduction

1.1 Background

Operational Support System Sales, OSS Sales, is a technical support unit within the Ericsson Corporation located in Gothenburg. OSS Sales handles sales related support issues for local Ericsson companies worldwide. The Ericsson Corporation consists of many local Ericsson companies worldwide. A typical request scenario could be if a local Ericsson company in, as an example, Turkey was engaged in selling telecom equipment supported by OSS Sales to a Turkish telecom operator. The Turkish operator could then be interested in technical details that the local Ericsson company is unable to answer. The local Ericsson company could then contact OSS Sales for support.

In spring 2002, a project was founded by OSS Sales at Ericsson. The scope of the project was to find a solution to the increasing perpetual stream of incoming sales support requests. From the beginning the main focus of this project was to create a whole new technical solution for the problem in shape of an own developed information system. Our role was to both design and develop a new information system. We made a requirement specification for this purpose according to an Ericsson requirement specification template. But after the telecom crisis turned out to be worse than expected the last nail in the coffin was put to place. No more expensive own developed systems for unique purposes unless necessary. The telecom happy days were now really over for this time. However, the OSS Sales still needed a new information system to make their sales support handling more effective.

Now the project focus turned to look more closely towards standard systems already existing within Ericsson for this purpose. The reason to only look for existing standard systems within Ericsson was mainly economical, but by choosing an already existing system there can also be benefits in internal knowledge and prior experiences with the system. After intensively searching the Ericsson intranet we found four suitable systems. The candidate systems were found after roughly comparing them to the requirement specification. A decision to make a more thorough comparison was made. Our main scope had now changed from developing a completely new system to account for the adequacy of these four systems for the OSS Sales organization. To do this we needed to perform an evaluation between these systems.

Now we encountered another problem. How does one evaluate standard systems? We searched the Internet and found a handful of existing methods for this purpose. Now we had reached a state where we had to make two choices instead of one. Firstly we had to choose a method for evaluation and then with the help of the evaluation method make a choice between the four systems. Another question we asked ourselves was, is there really a method fully suitable for our specific needs. We now felt a great deal of uncertainty to which we had to find an answer. We further came to the conclusion that there is most likely a need for supportive method to help organizations making sound choices.

1.2 Problem Presentation

Software evaluation practices used in organizations are for the most ad hoc and chaotic. The task of standard systems selection is often assigned under schedule pressure and as many organizations are just now increasing their standard systems usage, evaluators are often first timers at the task. They may not have the time or experience to plan the selection process in detail and, therefore, they may not use the most appropriate methods in the selection process. (Nilsson, 1991)

According to Andersen (1994), many organizations are under the delusion that choosing a standard system is a simple trivial procedure. They believe for instance that they are able to choose a standard system without really analysing and describing their own enterprise. It is common among organizations to believe that they could get away by choosing the cheapest or the most sold system. Andersen believes this to be fraught with danger and could have disastrous consequences. (Andersen, 1994)

Software systems do not exist in isolation, they are used in social and organizational contexts (Sommerville, 1995). Experience, and many studies, shows that the major cause of most software failures are people, rather than technical issues (Potts, 1993; Friedman and Kahn, 1994; Beynon-Davies, 1999). Even with the availability of a wide range of advanced software development models, methods, techniques and tools, serious problems with software are still being faced. Furthermore, when the selection process is not defined, it is reinvented each time, it is performed inconsistently and learning from previous cases is difficult. According to Nilsson (1991) a systematic approach can contribute to taking care of feasible opportunities when at the same time identifying and avoiding potential pitfalls.

We agree that choosing a standard system is more complex than just picking the first one that comes to mind. We also agree that a systematic approach can contribute to making a sound choice. But what is a systematic approach then? Nilsson has been involved in developing the SIV (Standardsystem I Verksamheter) method for software acquisitioning and that is an example of a systematic approach for choosing a standard system. Does that automatically mean that by, as an example, using SIV an organization can be sure of making a sound choice? What knowledge should the evaluation method provide and what knowledge is provided by available method for standard systems evaluation.

1.3 Purposes and Issues

Main purpose

The main purpose of this thesis is to contribute to an increased feeling of security and confidence in the process of making a sound choice of a standard system.

Sub purpose

- Create meta-architectural support for evaluating standard systems.
- Help OSS Sales to make a sound choice of a standard system for request handling.

Main issue

How should a meta-architectural support for evaluating standard systems be designed?

Sub issues

- What knowledge should the meta-architectural support provide?
- What knowledge is provided by available methodological support?
- Can this meta-architectural support be adequate for OSS Sales in helping them to make a sound choice of a request handling system?

1.4 Delimitations

We have delimited our thesis to only study the available knowledge in three existing methods for standard systems evaluation. Nevertheless, we have tried to distinguish these three methods as the foremost methods within the field. Further on, when applying our meta-architectural support in the case study at OSS Sales, we used a beforehand made requirement specification that was made according to Ericsson standards. The requirement specification does not embrace customer wants/needs, leaving our empirical case study with a delimitation regarding this part.

1.5 Disposition

The *first chapter* is a general introduction to this master thesis. This chapter consists of the thesis background, problem presentation, purposes and issues, and delimitations. Given the background it is vital to analyse the problem in order to find out what needs to be investigated in the thesis.

In the *second chapter* we present the chosen scientific methods used during this thesis. We also account for how material was collected and what material that was collected for this thesis and finally we bring out some critism to our choices.

The *third chapter* functions as a theoretical frame of references. The chapter starts out with basic descriptions on what a system is. Then we aim to create a greater understanding on the concept of what a method really is and how methods can be used in meta-modelling. The chapter finishes off with clarifying what knowledge is to be included in a meta-architecture in order to be able to make a sound and secure choice of a standard system.

In the *fourth chapter* we present our meta-architecture.

In the *fifth chapter* we apply the meta-architecture on a real situation at OSS Sales, Ericsson AB in Gothenburg.

In *chapter six* we analysed the results of our empirical work in terms of what knowledge we found in the chosen existing evaluation methods. We analysed what functional, structural and infological knowledge that was supported by each method.

In the *seventh chapter* we discuss the outcome of our thesis in terms of the original problem, our course of action and choice of scientific methods. We also discuss the result of the thesis and what insufficiencies that we are aware of and what could have been done otherwise. Furthermore, we also give suggestions on further research.

2 Method

This chapter aims to create an understanding of the scientific methods used to create this thesis. We also describe how and from where the material for thesis was collected and finally we will bring out som critism to our chosen methods and material.

2.1 Scientifical Methods

2.1.1 Qualitative & Quantitative Inception of Science

A qualitative inception is according to Backman (1998) based on a subjective viewing angle. In contrary to the traditional inception where one more or less observes a reality from an objective viewing angle the qualitative inception observes a reality from within the reality, thus meaning how individuals reflect on their surrounding based on their own perception. The qualitative inception is more focused to the individual than the traditional inception. (Backman, 1998)

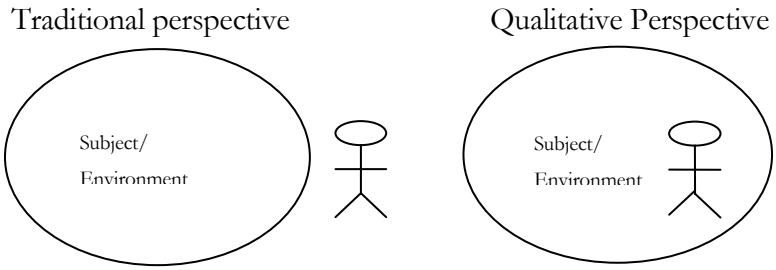

Figure 2.1 Traditional and qualitative perspective

Conspicuous notions within the qualitative inception are meaning, context and process. Meaning means observing how individuals experience, interpret and structure their reality in relation to their perceptional ability. In other words it means how to obtain meaningfulness in ones situation. Context is observing individuals in their natural environment and not in a laboratorial environment. (Backman, 1998)

Processes characterize the qualitative thinking more than products or results that constitute the traditional inception. Within the qualitative science it is not unusual that the scientist himself being part of the observed reality which contributes to a subject-to-subject relation. In contrary to the traditional science the subject-to-object relation is replaced with a realistic and perhaps even authentic subject-to-subject relation. (Backman, 1998)

A qualitative method aims to create a deeper understanding of the studied problem and its relation to its surroundings/environment. The knowledge achieved is not descriptive but hopefully clarifies and thus provides an understanding of the problem complexity. (Andersen, 1994)

The quantitative inception of science is more focused on data that can be converted into figures. However the quantification doesn't always have to be figures, if you say that for

instance, person A runs twice as fast as person B than that is also a form of quantification. Person A is then 100% faster than person B. But if you instead only say that person A is fast, then it's not a quantification. Quantified data can be processed statistically and can in most cases with great advantageous be presented in tables or diagrams to help the author in creating comparable information in somewhat small places. This also helps the reader to more quickly make own interpretations and assumptions. (Ejvegård, 1996)

2.1.2 Induction & Deduction

A qualitative inception is in most cases inductive. By inductive means in contrary to the traditional inception not to begin creating a base of knowledge from out of theories or hypothesis. That procedure is called deductive. When working inductive the direct focus is on the empirical part to later on formulate a hypothesis or a theory. (Backman, 1998)

According to Davidsson and Patel (1991), scientists who works inductive is on the discovering path while the deductive scientists follow the proofing path. The inductive inception focuses by researches to discover hypotheses or theories that is not already rooted in the conventional deductive inception. The deductive inception focuses on to prove conventional hypotheses or theories. (Davidsson and Patel, 1991)

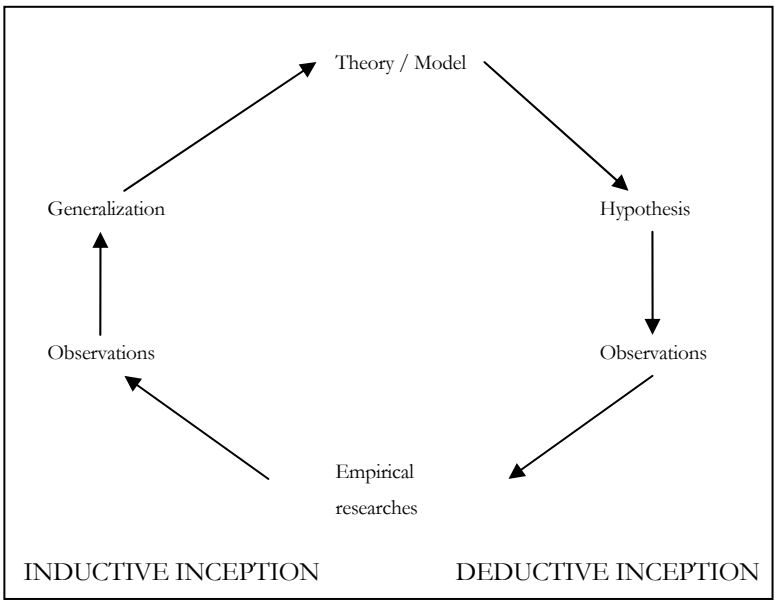

Figure 2.2 Inductive and deductive inception. (Source: Eriksson & Wiedersheim-Paul, 1999)

2.1.3 Professional Conversation Method

Ordinary conversation is probably not a formal recognized scientific method but rather an informal course of action. We are not aware of any particular scientific method that embraces our line of work, however there is a method called professional conversational method that pretty well describes our line of action. According to Zimsen (1998) there are certain characteristics that separate an ordinary conversation between friends to a professional conversation and that is:

- Person/ Persons in focus
- Target oriented conversation

- The use of well suited methods to reach a defined goal
- Purpose of conversation

2.2 Chosen Scientifical Methods

Since our study beeing based on seeing ourselves as the designers of a meta-architecture our empirical part in the creation of this consists of daily conversations with each other and once in a while with our supervisor. By this one can say that we are studying ourselves in our work. In that manner we would like to claim that we by our "closeness to the studied object" is working qualitative. The method who the most illustrates our daily conversations we have found to be the Professional conversation method.

By empircal work we come to a theory, or in this case a meta-architecture, that could help evaluating standard systems. By this we mean we have worked inductively. However later on we decided to empirically test this meta-architecture on OSS Sales and in that manner one could also say that we have worked a bit deductive.

2.3 The Collection of Material

Sources of litterature could be of both primary and secondarily nature. Material that is of primary nature mostly heritage from books, reports, dissertions, scientific articles and so on. Secondary material is material that has interpretated another source. The secondary source is therefore often not as reliable as the primary source. (Backman, 1998)

In our thesis we have to great extent focused on getting primary sources.

2.3.1 Collection of literature

Standard Literature

By standard literature we mean literature thats is most likely to be found at an ordinary library. We have searched the school library for books on method, methodologies and other related notions.

Specific Literature

By specific literature we mean litterature that is not likely to be found on a standard library but is still printed. To find this specific literature we have searched the Internet as well as the Ericsson Intranet. Other specific literature that we have not found on the Internet or Intranet has been a doctoral dissertion by Magulas & Pessi as well as the Delta report, which is a joint venture project between several organizations and the department of Informatics at Gothenburg Universtity. Specific articles that is hard to find elsewhere has been provided to us by our supervisor.

2.3.2 Personal Contacts

In cases when you can't find the information required elsewhere or if you want the information to come directly from the original source then you can contact the person responsible. In some cases during the information gathering on the available request handling systems within Ericsson we could'nt always find what we were looking for and therefore we were had to call or sometimes email the persons responsible to get answers to

our questions. We wouldn't really like to call these personal contacts interviews since we thought of them to be more of fast product related informational questions.

2.4 Critism of Chosen Scientifical Methods and the Collection of Material

2.4.1 Scientifical Methods

We see ourselves as two designers of a meta-architecture for supporting standard systems evaluation. However this study is based on our knowledge and what we felt was important. Round the world there are for sure thousands of designers that would have thought other aspects beeing of greater importance than the ones we present in this study. Therefore one could consider if a more quantitative inception would have been more successful in finding the best architecture for evaluation. However, we believe that would have been far beyond the range of a master thesis.

2.4.2 Collection of Material

The material we have chosen for this thesis is mainly material that heritages from universities, much of what we would like to call the core literature is heritated from the University of Gothenburg department of Informatics. Because of this we believe ourselves to have a good foundation from which to create our meta-architectural support upon. However since much of the foundational material heritages from one source the outcome might be of advantageous to that one source.

2.5 Reliability and Validity of the Study

Reliability relates to the trustworthyness of the study. Are the methods used for the study appropriate? Can the study be performed once more with mostly similar results? (Ejvegård, 1996)

Validity relates to if the researcher really is researching what is to be reasearched. As an example if you are to state how many people lives in America and is using the taxregister as the source of information then one could discuss the validity in that since the real population in America is approximately 5-20 millions more than what is registered. (Ejvegård, 1996)

2.5.1 Reliability & Validity

As mentioned earlier we believe that since we are only two designers in a world of many thousands perhaps even millions, we are aware of that the outcome of this thesis could have been different if the study was performed on more than us two.

We believe that we have used appropriate and trustworthy material during our study which we believe helps supporting the validity of this study.

3 Theoretical View

In this chapter, which reflects the theoretical view of our thesis, we will describe the theoretical knowledge from which we based our meta-architecture upon. The chapter starts with describing systems in general and in particular standard systems and request handling systems. The reason for this is because of the request handling systems that we are later on going to evaluate are within in the group of standard systems.

To further create an understanding on what basis we designed our meta-architectural support for our evaluation, we will describe methods and how they can be combined and how to look at the concept of method from a knowledge perspective.

We then present theories on what characterizes a successful information system and what principles that determines its "goodness".

We also look at available knowledge in existing methods for standard system evaluation.

3.1 Concepts of System

3.1.1 What is understood by Systems in general?

The concept of system is widely used and can be used in a variety of science fields. The most fundamental description of what a system is refers to the system being a collection of parts related to each other. There are no limitations in what the parts of the system consist of; also there are no limitations in how they are related to each other. A system can consist of mostly anything from the tiniest cell in a living organism to the solar system itself. A system doesn't have to be physical, it can also be non-physical as the social systems. (Andersen, 1994)

The relations within the system form the activity of the system, which in turn creates the whole of the system. The wholeness lifts the system up to another level of analysis, which means that the sum of all parts and their relations creates an outcome greater than the sum of all parts accounted separately. As long as the relating parts in the systems create this wholeness, the system is stable. If a part of the system is taken away the relations in the system can be disordered causing the system to be unstable which in turn affects the wholeness of the system. (Joslyn, 1993)

3.1.2 What is understood by Information Systems?

An information system is a system made by humans for humans. The information system is tied to a certain task where its main purpose usually is to pass on information from one user to another. The information system can furthermore perform actions based on defined rules in the information system leading the information system to present transformed information to the end user. (Andersen, 1994)

Andersen's (1994) formulation of a definition for an information system is that *"an information system is a system for gathering, processing, storing, transferring and presenting information"*.

Buckingham et al (1987) defines an information system as *"a system which assembles, stores, processes and delivers information relevant to an organization (or to society), in such a way that the information is accessible and useful to those who wish to use it, including managers, staff, clients and citizens. An information system is a human activity (social) system which may or may not involve the use of computer systems".*

3.1.3 What is understood by Standard Systems?

Standard systems are systems that are developed to fit not only one unique organization but many. Therefore a standard system is more generalized than a tailor-made system. However, purchasing a standard system does not always equal sacrifices in functionality. Many standard systems can be more or less modified to better fit the organizational needs. It is also very common among standard system developers to have a portfolio of several different applications for the customer to chose/add from. (Andersen, 1994)

Classification of Standard Systems

Since some standard systems can be more or less modified than other standard systems it could be of certain value to further classify the concept of standard systems. Figure 3.1 shows a classification of standard systems based on level of standardization. (Nilsson, 1991)

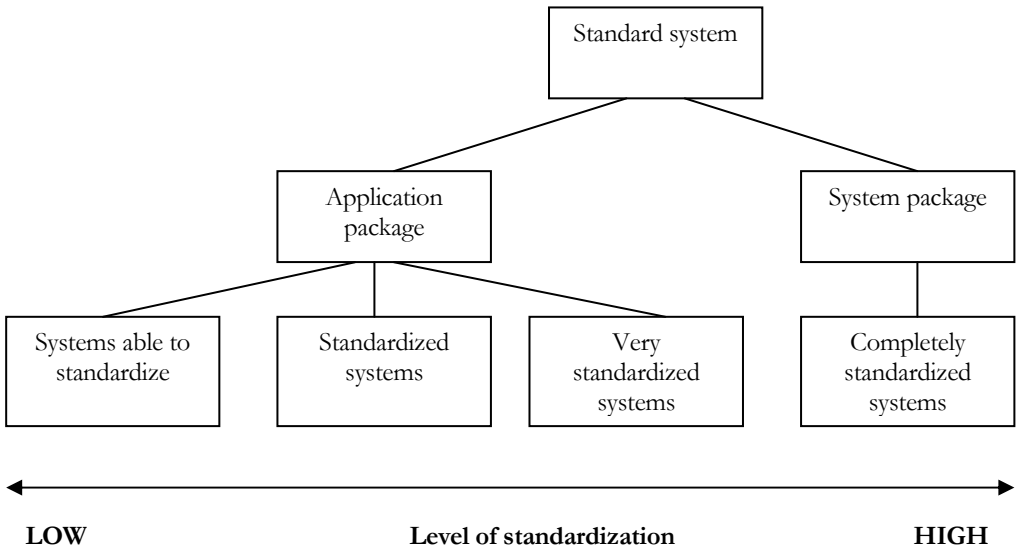

Figure 3.1 Classification of standard systems. (Source: Nilsson, 1991)

System able to standardize

A system that is able to standardize is a system where the supplier defines a frame for the application area from which the customers can develop their own system. The customer is to say provided with a platform with high level of flexibility. (Ibid)

Standardized systems

A system that is standardized is a system where the supplier has a more standardized application suite. The supplier can for instance provide a hard coded source for the system from which the customer can make modifications to fit the organization. (Ibid)

Very standardized systems

Very standardized systems are even more standardized. Only small modifications are possible. (Ibid)

Completely standardized systems

Completely standardized systems are most often independent systems that come in a package ready to install. (Ibid)

3.1.4 What is understood by Request Handling Systems?

In a request handling system there are descriptions of working procedures inside an organization. These working procedures are described as requests inside the request handling system. Some procedures in an organization would most likely benefit from being computerized like for instance the process of requests handling. Most types of requests has a more or less sequential path between coworkers in or between organizations and with the help of using a request handling system one can more easily and rapidly make a follow up and check the current status of the request. A typical request could be as an example someone who needs to know if product A is compatible with product B. (Arilla, 1999)

The request handling systems helps the handler of the request in his or her daily work by making the work much more effective. It is also a benefit for the requester who can receive fast answers regarding his or her errand. The system can furthermore assist the handler in many areas. The system can for instance send out a warning if a request seems to be forgotten or is near due-date. Furthermore there are advantages in using a request handling system for the manager him/herself as well as many systems has a built in or add on statistical function. The statistical function can be useful in many areas like presenting data on how many request is coming in, how many requests are from a certain requester, what do the requesters mainly want to know, are the handlers doing a good job and so on. (Ibid)

Key characteristic features for a request handling system are surveillance, logging and controlling the request status. (Wessbrandt, 2002)

3.2 Concepts of Method

The concept of method has been discussed for some time. System development methods are often used during system development in order to guide and support the system development process. However, the concept of method is not always clear. When studying different theories about methods or looking into different practical manuals it is obvious that several definitions of the concept of method exist. Furthermore, there exist several related concepts, often with "method" as a prefix and a following noun (method alliances, method components, method fragments). When looking more closely at different concepts, one can identify that there are different concepts and terms used for the same phenomenon and also the same concept and term for different phenomenon. We will in the following text in this chapter try to further explain these varieties of concepts and terms to hopefully create a greater understanding of the word method in our situation.

3.2.1 What is understood by Method in general?

Jayaratna (1994) defines "method" as *"an explicit way of structuring one's thinking and actions. Methodologies contain model(s) and reflect particular perspectives of 'reality', based on a set of philosophical paradigms. A methodology should tell you 'what' steps to take and 'how' to perform those steps but most importantly the reasons 'why' those steps should be taken, in a particular order."*
As we can notice, Jayaratna uses the term methodology.

Methodology is a Greek term meaning the study of methods. The Oxford Dictionary defines methodology as *"the study of systematic methods of scientific research"*.

3.2.2 What is understood by Method in Information Systems Development?

Jayaratna justifies the use of methodology claiming that *"the term methodology is pragmatically well established within the field of information systems to mean the same as method"*.

Another example of the use of "methodology" synonymously with "method" is that of Stamper (1988) when stating: *"I use the term 'methodology' under protest bowing only to customary usage. It would be better, as in the philosophy of science, to speak of 'methods' when referring to specific ways of approaching and solving problems, and to reserve methodology' for comparative and critical study of methods in general; otherwise this vital field of study is nameless"*.

Further on, Brinkkemper (1996) states: *"the misuse of the term methodology standing for method is a sign of the immaturity of our field, and should consequently be abandoned"*. Brinkkemper defines method as an *"approach to performing systems development projects, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products"*.

Röstlinger & Goldkuhl (1994) have a third definition of the concept of method when saying that *"methods are prescriptions for human actions and methods are normative and guide the Information Systems Development process"*.

Checkland (1985) distinguishes method and methodology by saying that *"a methodology is a set of principles of method, which in any particular situation has to be reduced to a method uniquely suited to that particular situation"*. Another definition from Checkland (1981) is that *"a methodology will lack the precision of a technique but will be a firmer guide to action than a philosophy. Where a technique tells you 'how' and a philosophy tells you 'what', a methodology will contain elements of both 'what' and 'how'"*. Checkland also uses the concept of technique but what he exactly means by technique is not defined.

Andersen (1994) defines a method like a *"detailed description of course of action for solving a certain problem. The method is characterized that it has a field of application, which shows what type of problems the method can be applied to"*.

Avison and Fitzgerald (1995) also define a method like a gathering of course of action, techniques, tools and support for documentation. They mean that a real method takes its point of departure in a philological outlook and is more then just the terms mentioned above. They also mean that methods for developing information systems are about balancing knowledge about technology with knowledge about human behavioural pattern.

When we examine the method definitions above, it is clear so far that the term "methodology" is often used when what is actually referred to is "method". Method is

descended from the Greek language, meaning "way of investigation". The meaning of "method" seems to answer the question of how information systems development shall be performed.

3.2.3 Other related notions

Method Types

Nilsson (1991) presents the concept of "method type" (in Swedish: metodik). He distinguishes between a method type and a method when he defines a method type as a general concept (a type of method) and a method as a specific concept (an instance). In other words a method is a concretion of a method type.

Method Chains and Alliances

Fåhraeus (1986) talks about "method chains" as a consisting of several methods linked to each other. Further, the result from a method used in an earlier step shall be used in a following step.

Nilsson (1998) has further developed the concept of method chain and defines it like *"integration of methods between different levels of development work. This approach to combining methods is a kind of a vertical integration"*. Nilsson points out that there are several abstraction levels of development work. A method alliance is an integration of methods within the same level of abstraction. This is a horizontal integration of methods. Nilsson states that alliances are motivated by the need "*to tackle several problems or perspectives in concrete situations*". That is, method alliances cover several aspects of a problem domain at a specific level.

Model/Framework

Another related and sometimes confusing term is "model". According to Yourdon (1989), a model is used to "*highlight, or emphasize, certain critical features of a system, while simultaneously de-emphasizing other aspects of the system"*.

Rumbaugh et al. (1991) define a model as *"an abstraction of something for the purpose of understanding it before building it"*.

Jayaratna (1994) defines "framework" as a static model that provides a structure to help connect a set of models or concepts.

Goldkuhl (1991) defines "model" as a structure for the information systems development process. Further, a model defines and delimits specific areas within information systems development that form related phases. A model answers the question of *what* is to be done but not *how* it should be done.

What makes the definitions above confusing is that they are referring to different domains. When we examine Yourdon's definition, it is obvious that he is referring to a model of an information system. The same goes for Rumbaugh et al. However, when Goldkuhl talks about models he is referring to a model of the information systems development process.
In other words, they are using the same term but referring to different concepts. Jayaratna (1994), similarly to Goldkuhl, refers to the information systems development process whilst the others refer to the product of such a process.

In Röstlinger & Goldkuhl (1994), the framework concept is also used as a synonym to model. The concept "framework" is well defined in the software engineering community but not fully applicable in the information system community.

Öberg (1998) gives one definition from the software engineering community. He defines the concept of framework as *"a generic design solution to a certain problem or a certain domain. The framework describes the different design elements involved in the solution, as well as their relations"*.

If one changes the term "design solution" to "ways of performing information systems development" and the term "design elements" to "phases" the definition becomes similar to Röstlinger & Goldkuhl's (1994) definition of framework/model.

Perspective

Another method-related concept is "perspective." A perspective is a theory of how information systems development shall be performed (Nurminen, 1988). This theory shall be normative, explanative and classifying.

Mathiasen (1982) defines perspective as a conceptual abstraction of a view or a specific phenomenon.

Jayaratna (1994) says, *"methodologies ... reflect particular perspectives of 'reality' based on a set of philosophical paradigms"*.

In other words, the method designers' perspective is based on how he or she perceives the world. The method designers' values and beliefs influence the system developer when performing information systems development. A perspective implies, for example, what primitives to use and these primitives in turn influence method users (i.e. system developers). The character of the influence can be either governed or supported. The perspective is not necessarily made explicit in the method. The method designers' perspective is often implicit and taken for granted. One can say that a method is always based on a perspective from which follow (Goldkuhl, 1991):

- Principles
- Values
- Conceptions
- Experiences
- Categories
- Definitions

We can distinguish between internal and externalized perspectives of a method creator (or any human being). The internal perspective is constituted by the parts of the conception of the world that are hard (or even impossible) to externalize. The externalized perspective, on the other hand, is constituted by inter-subjective beliefs etc., to which the method creator adheres. Examples of existing externalized perspectives in information systems development are business-orientation, object-orientation and user-centered development. To sum up, the perspective, explicit or implicit, influences the method user in one way or another.

Method Components and Method Fragments

A method can be perceived as a "whole" consisting of different "parts". Therefore we also need a concept for the parts of a method. Lately, concepts such as "method components" (Röstlinger & Goldkuhl, 1994) and "method fragments" (Harmsen, 1997) have been proposed to discuss method parts. A reason for this is a move from viewing methods as monoliths to a generic flexibility (Röstlinger & Goldkuhl, 1994) suited for situational method engineering (Harmsen, 1998; Brinkkemper et al., 1998).

The concept "method fragment" is defined by Harmsen (1997) as *"… a description of an information systems engineering method, or any coherent part thereof"*.

To sort this out, a method fragment is said to reside on a certain layer of granularity, of which five are possible: method, stage, model, diagram or concept. Furthermore, a method fragment is either a process fragment or a product fragment. Process fragments represent the activities, stages, etc., that are to be carried out and product fragments represent deliverables, diagrams, etc., that are to be produced, or that are required during development.

Röstlinger & Goldkuhl (1994) view methods as constituted by exchangeable and reusable components. Each component consists of descriptions for ways of working (a process), notations and concepts. A process describes rules and recommendations for the information systems development and informs the method (component) user what actions to perform and in what order. Notation means semantic, syntactic and symbolic rules for documentation. Concepts are categories included in the process and the notation. A method component can be part of a method chain or a method alliance. A method component or fragment can also be used separately and independently from other components. Each method component addresses a certain aspect of the problem at hand and is part in a whole (a method). Therefore, a method component can be thought of as the smallest meaningful assembly of method fragments to address a certain aspect of a problem (cf. Brinkkemper et al., 1998) and consists of product fragments (notation), process fragments (process) and concept fragments (concepts) used in the other two types of fragments. Note that a method component per se is a method fragment at some intermediate layer of granularity.

Cooperation Forms

The Scandinavian tradition of performing information systems development often means that several actors are involved in the information systems development process.

Hägerfors (1994) describes the information systems development process as a group process with actors who interact, discuss, learn, agree, disagree and argue. Several research reports argue for strong user (business actor) participation. This means that methods also should support cooperation forms.

According to Goldkuhl et al. (1997), cooperation forms describe *"how different persons interact and cooperate when performing method-guided work"*. Cooperation also has to do with roles and division of work. One can say that co-operation forms deal with the meta-question of who is to ask the questions during information systems development. Examples of cooperation forms are brainstorming sessions, interviews and modeling sessions. (*Ibid.*).

Harmsen (1997) distinguishes between two different domains that are in focus during

information systems development. Some information systems development activities belong to the "target domain" and some to the "project domain". The target domain consists of activities directly addressing information systems development, and the project domain consists of activities addressing management thereof. Cooperation forms thus belong to the project domain.

3.3 The Concept of Methodological Knowledge

3.3.1 What is understood by Method from a Knowledge Perspective?

Different design methods are more or less complete regarding methodological knowledge. Therefore, different design methods are more or less suited to reduce insecurity in the development work. (Bergenstjerna et al., 1999)

Seeing that a design method is built upon an amount of methodological knowledge, we chose to classify methods from a knowledge perspective. The classification of design methods aims to clarify what type of methodological knowledge that is available and what type of methodological knowledge the different methods embrace. (Avison & Fitzgerald, 1995)

A design method offers methodological knowledge for issues concerning quality, decision of change, process of change (i.e. procedural knowledge), as well as people, information environment and IT (i.e. substantive knowledge). Through that the design method contributes to increased comprehensibility. Further on, a design method offers the methodological knowledge needed for handling issues concerning experience (i.e. descriptive knowledge) as well as wanted future circumstances (i.e. normative knowledge). Through that the design method creates conditions for understandability (awareness), which leads to understanding and acceptance of a larger whole. (Bergenstjerna et al., 1999)

Issues regarding management of change		
Management of quality	Management of decision of change	Management of process of change

Figure 3.2 Procedural issues. (Source: Bergenstjerna et al, 1999)

Issues regarding relations within an IS- enviromnament i.e.
Relations between IT, humans, processes and structures

Figure 3.3 Substantive issues. (Source: Bergenstjerna et al, 1999)

A procedural and descriptive design method balances knowledge on how issues concerning the design process should be handled, as well as how existing circumstances and experiences of these issues should be trapped. A procedural and normative design method balances knowledge on how issues concerning the design process should be handled, as well as how visions about future circumstances regarding these issues should be trapped. (Ibid)

Substantive and descriptive design method balances knowledge on how issues concerning the design product should be handled, as well as how existing circumstances and experiences of these issues should be trapped. A substantive and normative design method balances knowledge on how issues concerning the design product should be handled as well as how visions of future circumstances regarding these issues should be trapped. (Ibid)

Hoffman (1998) points out the importance of substantive and procedural knowledge by saying that individual, organizational and social expectations or dreams cannot be materialized without adequate meta-architectural support, i.e. ways to organize the existing substantive and procedural knowledge.

3.4 Design of Meta-Architectural Support

Design is a means for supporting comprehensibility and understanding. It may be seen as a desideratum for sound information systems development. This last section provides a tentative view of the unbalance between required and existing methodological knowledge for the support of information systems development.

3.4.1 The Design Situation

In this section we will show upon how we as the designers of an organizational change function in this whole process. To illustrate this we will use Peter Checklands Soft System Methodology for the management of continuous development since it provides the proper context for understanding the whole process of development and it refers to a proactive philosophy of managing a continuous development rather then a reactive one.

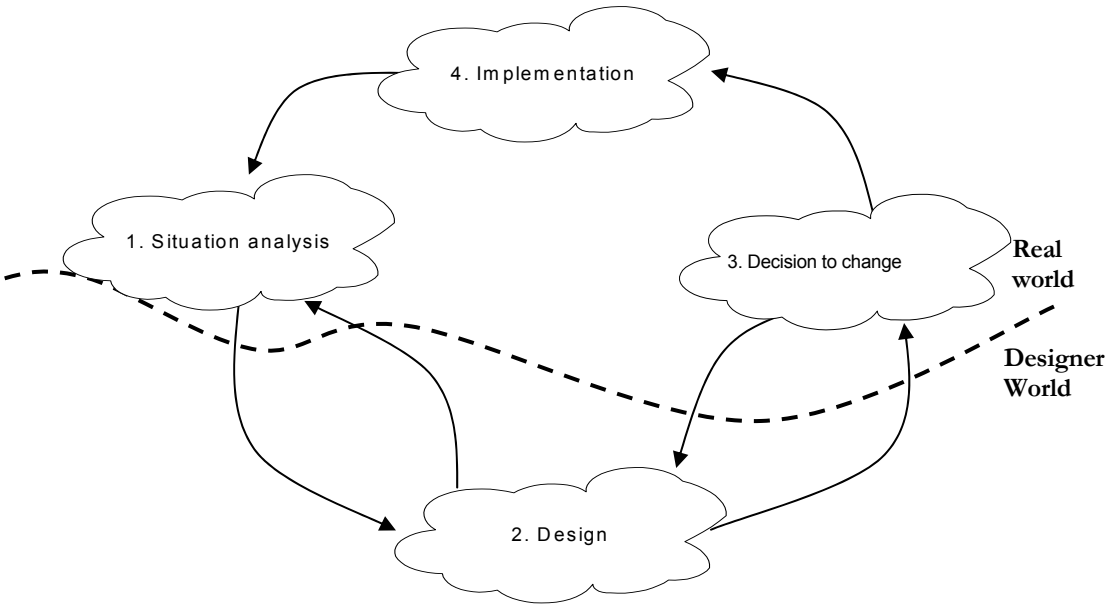

Figure 3.4 The learning model

Sometimes organizations know what they want, sometimes they believe them self to know what they want and in some cases they do not know what they want at all. In either case the designer must establish good communication with the organization to get the whole picture on how the organization work and how they want and need to work in the future to be as efficient as possible. One can say that the "inhabitants" in the two "worlds", illustrated in figure 3.4, are both professionals but in different areas. The real world consists of the organizational and the people involved in it. The people involved could be anyone from managers to customers. However they are experts on how their world function and hopefully they have a notion on what their goals and objectives are with the new system. (Checkland, 1990)

The Real World

The real world in the Checkland figure above reflects the organization in need of change. If there is an existing system or working strategy implemented that is not working perfectly well in one way or another, a need for improvement is awoken. This is illustrated in step 4-1. The designers have to find out *what* the problem is and *how* to deal with it. Organizations are often very complex due to the many different individuals of which the organization consists of. Individuals are autonomous and have their own interests, needs and desiderata. Therefore Checklands advocates that the interaction between humans within in an organization is very important for the wellbeing of the organization. The same goes for system or organizational improvement. If communication is limited to only a part of the organization the risk of misunderstandings is overwhelming as well as the system implementation not being as successful as wanted. Checkland therefore claims the involvement of every part affected in the change to be very important. According to Checkland communication represents a major part of the learning process. (Ibid)

The Designer World

The designers of the designer world are professionals in both analysing and designing the real world. The designer must also have a certain amount of psychological skills in case the people of the real world are uncertain of what they want or if they know what they want but can't express it in simple words. The designers usually also provide consultation on the whole process by making sure process quality is achieved regarding human, social, organizational and systemic issues. The designers strive to achieve a comprehensive picture. (Ibid)

The designer world is to say in a meta-reality where the designer must think about the real world in systemic terms. The communication between the real world and the designer world is as mentioned above very important and it is almost impossible to only have one or only a few times of communication with the organization. Instead the communication needs to be an iterative process as illustrated in step 1-2. (Ibid)

The Learning Process

The learning process between the real world and the designer world is an iterative process where the real world learn the designer about the situation by answering questions asked by the designer as they occur throughout the process. From each question asked the designer gets an even better knowledge of the real world helping the designer to understand the organization from both a descriptive and a normative perspective. From the gained knowledge the designers can make a proposal for a change, illustrated in step 2-3 in the

figure above, and if accepted implement the change. Most of the time the proposal isn't fully accepted at the first time and smaller changes has to be made. Therefore Checkland claims step 2-3 to be iterative and a learning procedure as well. Actually, since most organizations find them self in a continuous evolvement the whole figure is iterative. (Ibid)

Cultural & Political Issues

Another important issue according to Checkland are aspects of cultural and political nature. According to Checkland there is always cultural and political aspects involved in any organization during almost any decision of greater scale. The cultural issues reflect social values and norms within the organization and between the organization and its surrounding world. The political aspects can be of intern as well as external nature and these kinds of political issues do not imply deep discussions of traditional nature. Political issues according to Checkland involve any area of conflict involving humans and their differences in interests, opinions or beliefs. (Ibid)

We believe cultural and political issues to be very complicated to further explain and sort out and therefore we have delimited our self not to go any deeper within that area in our thesis. However, we brought it up because we believe it to be an important aspect to be aware of.

3.4.2 The Designers Need for Knowledge

Security in development work

Applied in a situation, the meta-architectural support is expected to give answers to important questions regarding the substantive nature of the enterprise (know-what orientation) and the processes that change or modify it (know-how orientation).

Methodological Knowledge	Descriptive Knowledge	Normative Knowledge
Substantive Knowledge	Know-What to secure good product quality	Know-What to secure good product quality
Procedural Knowledge	Know-How to secure good Process efficiency	Know-How to secure good Process efficiency

Figure 3.5 Taxonomy of Methodological Knowledge. (Source: Bergenstjerna, 2002)

In a given situation, substantive issues expresses management of quality regarding the essential elements in an information environment like IT, people, processes and structures, business environments, and the mutual or inhered inter-dependencies that link these elements together. The substantive issues concern, for instance, informational aspects between information systems i.e. systems coordination and systems cooperation. (Bergenstjerna, 2002)

Procedural issues express the management of change i.e. the continuing development and management of quality, change decisions and change processes. The procedural issues are, for instance, questions of information technology use, of enterprise-wide development, of systems development, of knowledge development as well as the rate of change. (Bergenstjerna, 2002)

Quality in Information System

Magoulas and Pessi (1998) describe what characterizes the success of an information system. They presuppose from *functional*, *structural* and *infological* relations, where the relations shows the relation among the technology and the building stones of the enterprise. The relations should be characterized by harmony to mutual contribute to success.

According to Magoulas and Pessi (1998), the relations are characterized by qualities that express people's judgment and valuation of an information system. To form an opinion of the quality in the information system, identified qualities can be evaluated.

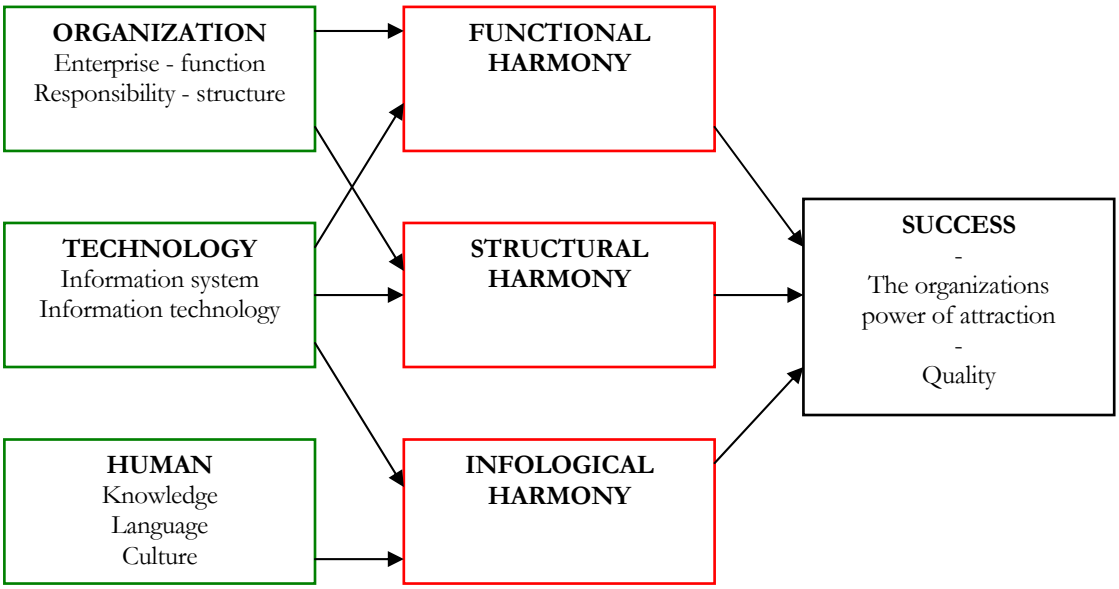

Figure 3.6 Functional, structural and infological harmony together contribute to organizational success. (Source: Högberg, 2000)

Functional Harmony

Functional relations refer to the reciprocal relations that appear among the action of people (i.e. functions and structures of the organization), in relation to the studied information system. The relations are expressed differently, depending on which enterprise that is represented. In many cases groups of actors are sharing the same information, which in the functional relation constitute a critical resource. This leads to special patterns of maintenance of information that should fulfill special quality and accessibility requirements. A well-shaped functional architecture is expected to achieve a balance between the information systems need of knowledge and accessible resources of information. (Hage, Kerim and Zamirian, 2002)

Functional harmony aims to create understanding on how information environments change as a consequence of technological and functional change. A fundamental condition for increased understanding of qualities, complexity and dynamics of an information environment, is to create comprehensibility. The purpose with functional harmony is to create a balance between copiousness of variation and social security. (Ibid)

The principle of indivisibility implies that the form of information-maintenance shall be in harmony with the form of the enterprise. The judgment of which form of integration that should be selected is done from human awareness and own experience. Such a principle creates motivation and simultaneously reduces the risks for information-islands and information-labyrinths. This makes, co-operation of systems and comprehensibility, possible. (Ibid)

Functional suitable design promotes creativity, freedom of action, dynamic problem solving etc. To sum up, the functional harmony can be characterized by qualities like functionality, economy, flexibility and efficiency etc.

Structural Harmony

Structural relations concerns issues on internal relations regarding responsibility, power and ownership. Since an individual objective in an organization hardly are socially universal but instead represents interests reflecting individual power the information system environment will be affected. (Ibid)

Structural harmony aims to clarify the organizational dispersal in order to create a ground for balance between power and responsibility. By doing this the structural harmony at the same time reflects upon how humans on one hand aims to create resistance and chaos and on the other hand willingness, openness and motivation. States of disequilibrium in the structural harmony creates incongruity between liberty and order and in some cases even uncertainty regarding ownership. (Ibid)

All information systems in an information environment have a field of responsibility where harmony between field of activity, knowledge and responsibility must reign. Terms of ownership, responsibility and motivation are essential for the structural perspective architecture. By this one can say that the structural harmony ensures the power and responsibility over the organization to be indivisible. (Ibid)

Infological Harmony

The infological view is built upon relations among people, their aims, visions and expectations. Every human is unique. To be able to belong to a social group, one human must accept the group's pattern of action, interpretation and communication. (Ibid)

The infological relations focuses on the individuals and the cultural, course of action, way of interpretation and communicational norms that exists within the organization. The infological relations therefore cover the states of dependences that are needed for communication. Infological harmony aims to give understanding for the upcoming balance between individual freedom and social responsibility for development and success of the wholeness. (Ibid)

The aspects of design are to adapt information systems to the individual's language, experience, etc. The infological harmony is characterized by qualities as intelligibility, knowledge, motivation etc. (Ibid)

Functional, structural and infological relations together lay the foundation of the concept of information, which is information as a resource, information as a force and information as a contribution of knowledge. The purpose with clarifying the qualities is to create

comprehensibility, understandability (awareness) and meaningfulness in information systems. (Ibid)

Hage, Kerim and Zamirian (2002) have with the help of Magoulas and Pessi (1998) identified important qualities and relating questions in an information environment.

Functional Relations
Qualities and questions support the understanding of people's comprehensibility in a system.

Qualities	Questions
Functionality	How does the system-functions support the enterprise?
Accessibility	How accessible are the functions in the system?
Efficiency	How effective are the functions in the system?
Flexibility	How easy is it to change functions in the system?
Stability	How stable is the system?
Economy	How has finance influenced the system development? How has the system influenced the enterprise economy?

Structural Relations
Qualities and questions on issues that supports the understandability and awareness of a system.

Qualities	Questions
Rights and responsibility	How are rights and responsibility issues allocated within the system?
Security	How safe is the system regarding delicate information?

Infological Relations
Qualities and questions supporting the understandability on how meaningful people experience the system to be.

Qualities	Questions
Intelligibility	How is the system language suited for the user? I.e. How user friendly is the system?
Relevance	How relevant are the tasks for the system users?
Knowledge	How easy is it to find information on the system?
Competence	How high is the system users knowledge on the functions in the system?
Motivation	How motivated is the users to use the system?

3.4.3 Knowledge Available to The Designers

In this chapter we will account for the knowledge available regarding three of the most commonly used methods for standard system evaluation. We have chosen to look more closely to them from the perspective of *comprehensibility, understandability (awareness)* and *meaningfulness*.

Anders G. Nilsson (1991) has chosen to classify methods for evaluating standard systems from out of application width, range of method, standard systems vs. hardware and standard systems vs. supplier.

We have found these classifications to comprehend quite well with our perspectives and therefore we will try to classify the three evaluation methods from out of Nilsson's classifications.

Method Application Width

By method application width Nilsson means how many areas of applications that are in range of the method.

- *Specific method:* A method that supports only a specific area

- *General method:* A method that is more comprehensible within most application areas. A general method contains mutual guidelines for obtaining different kinds of standard systems.

- *Mutual method:* A method that contains both a comprehensive part and a part with the possibility to make a more intense investigation within one or more specific application area. A mutual method can be a combination of a general method and a number of specific methods.

Method Range

By method range Nilsson means how comprehensive the method itself is within in the field of working-steps and types of documents.

- *Minimal method***:** Contains only the most essential elements for more simple problem situations. Using the minimal method as starting-point add-ons can be made to make it suitable for a specific situation.

- *Normal method:* Takes the starting-point in an average situation for a standard system evaluation. With this method as a starting-point you can make add-ons or reductions to the method.

- *Maximal method:* Is or should be covering all aspects and is feasible for many situations that may occur doing practical work. From the maximal method it is possible to derive fast acting variants of the method.

Standard Systems vs. Hardware

Nilsson wants to emphasize the method regarding its relation between the standard system itself and the required hardware.

- *The freestanding method:* This method implies looking at the standard system itself and does not pay any attention at all to the required hardware.

- *The co-ordinated method:* This method implies having a primary focus on the standard system but in contrary to the freestanding method the co-ordinated method takes some consideration in hardware issues. The hardware issues can function as an underlay if it is possible at all to implement the standard system on existing hardware or if new hardware has to be purchased.

- *The complete method:* This method contains integrated sections for choosing standard systems and the hardware to go with it.

Standard Systems vs. Supplier

Here Nilsson wants to emphasize on the method regarding its relation between the standard system itself and the supplier of the standard system.

- *Separate method:* A separate method in this situation implies the method to delimit itself to only study the standard system itself. The signification of the supplier is toned down.

- *Overall method:* An overall method emphasizes on making an overall judgement on the standard system and its supplier. The method tries to find the best combination between standard system and its supplier.

Standardsystem I Verksamheter (SIV)

The applications applicable with the SIV method for acquisitioning a standard system support the *general method*. This means that the method can be used in most cases when acquisitioning a new standard system in an organization. The figure below shows where the main focus of the SIV method is in its whole working area.

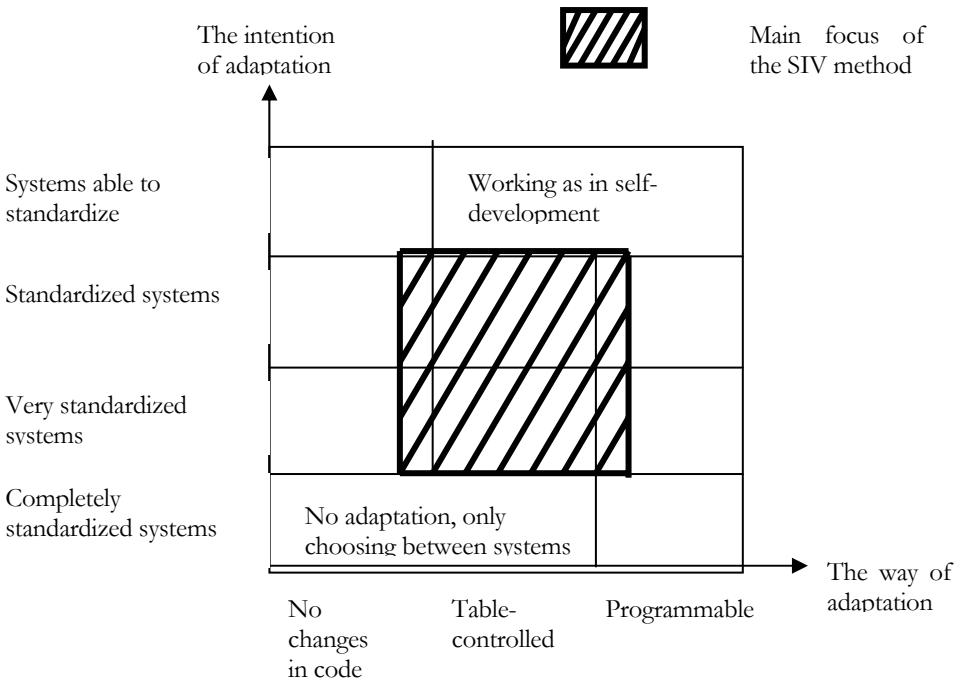

Figure 3.7 Main competence area of the SIV method.

The SIV method is originally a complete method for the whole process of acquisitioning a standard system. It is therefore not only a method for evaluation between different systems even though its main focus circles around making a sound choice. The method consists of extensive descriptions on the whole process of acquisitioning from the beginning to the end including both working-steps and templates for documents that needs to be written. This contributes to the classification of the SIV method as a *maximal method* regarding its range.

Regarding SIV and how it relates to software and hardware issues the SIV method follows the *coordinated method*. This origin from the SIV method developer's beliefs in not paying too much attention to the hardware since it could come to dominate the software issues. Nevertheless they believed that one could not completely neglect the hardware issues since they still regarded them as an important part of the acquisitioning process and that they could come to decide the matter in an evaluation process.

When it comes to SIV and how it relates to software and supplier issues the SIV method follows the *overall method*. When developing SIV the developers found that when choosing a standard system you are also choosing a supplier. To feel confident and secure with your supplier in areas of support, guarantees and other agreements they believed important for a successful implementation.

Functional Relations

Qualities and questions supporting the understandability of people's comprehensibility in a system.

Qualities ***Questions***

Functionality: SIV has a checklist that includes system functionality and exhorts the designer to list all functions in the standard system and compare them to the requirements in the requirement specification. The requirements should be weighted according to relevance.

Accessibility: The SIV checklist has a few pointers where it brings out issues like usufruct, services and so on.

Efficiency: In the checklist there are issues like system performance and access times to answer questions of efficiency.

Flexibility: The SIV checklist has one part that focuses on system flexibility and extendabilities in order to meet the changing needs of the enterprise.

Stability: The SIV checklist takes notice in the systems fault-frequency.

Economy: The checklist provides an economical judgement base for purchase, adjustment, maintenance and system updates.

Structural Relations

Qualities and questions on issues supporting the understandability and awareness of a system.

Qualities ***Questions***

Rights and responsibility: The SIV checklist doesn't include issues regarding rights and responsibility within the system. However SIV brings out more all-embracing issues between the supplier and the enterprise.

Security: The SIV checklist has a part that focuses on security issues like data-security and authority's.

Infological Relations

Qualities and questions supporting the understandability on how meaningful people experience the system to be.

Qualities ***Questions***

Intelligibility: The SIV checklist has a part that focuses on what language is used in the menus and user manuals. It also has another part that focuses on system structure, graphical layouts

Relevance: SIV has a part that focuses on how usable respectively redundant functions are within the system.

Knowledge: SIV has a few areas that focus on supplier competence, user training and what documentation is available.

Competence: Not supported within the SIV method.

Motivation: Not supported within the SIV method.

IEEE Recommended Practice for Software Acquisition

This is a recommended practice for performing software acquisitions. In this recommended practice, software products have been classified according to the degree to which the acquirer may specify the features of the software: commercial-off-the-shelf (COTS), modified-of-the-shelf (MOTS), and fully developed software. COTS are commercial software that is normally well defined and stable in its functionality. COTS products are not likely to be modified for a specific customer and can therefore be classified within *completely standardized systems*. MOTS on the other hand, is software products that you or the software company are able to modify to fit a special customers requirements, and can therefore be classified within *standardized or very standardized systems*. Fully developed software is software developed for a unique purpose and a unique system, and can therefore not be classified within standard systems at all.

This recommended practice can be applied to software that runs on any computer system regardless of the size, complexity, or criticality of the software. However, this recommended practice is more suited for use on modified-off-the-shelf software and fully developed software. As mentioned above, fully developed software cannot be classified within standard systems and by that means doesn't fit within the frames for this master thesis.

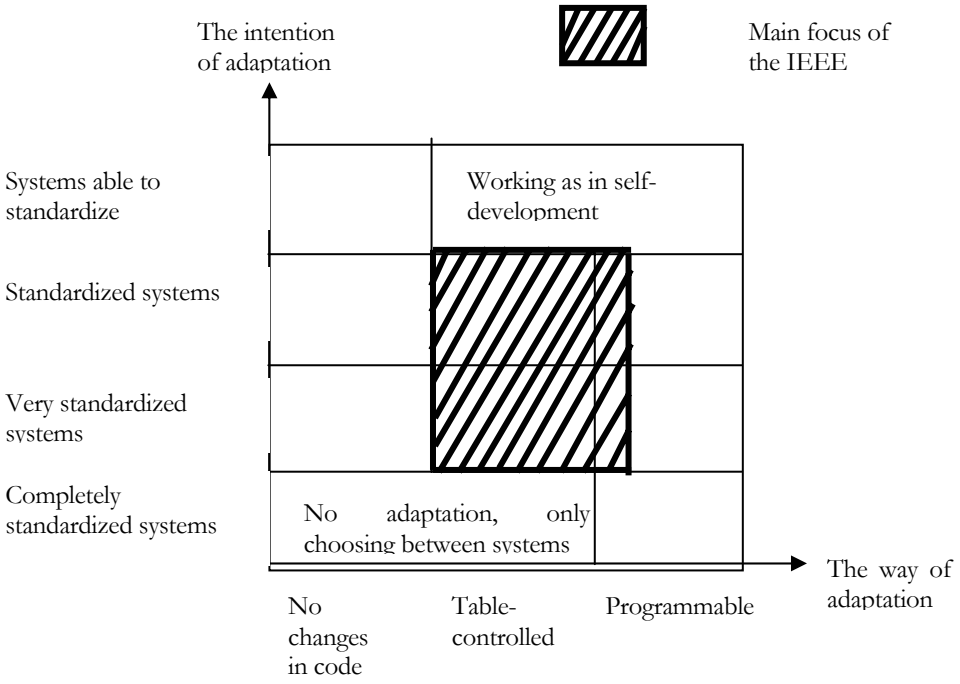

Figure 3.8 Main competence area of the IEEE Recommended Practice for Software Acquisition.

IEEE Recommended practice for software acquisition is comprehensive and can be used within all conceivable application areas. It also includes mutual guidelines for acquisition of different standard systems. This recommended practice could, by that means, be said to be a *general method.*

The IEEE practice consists of a set of useful quality practices that can be selected and applied during one or more steps in a software acquisition process. This recommended practice also embrace a comprehensive description of the acquisition process and its included working-steps and checklists. This recommended practice could, by that means, be said to be a *maximal method.*

Regarding the IEEE practice and how it relates to software and hardware issues the framework follows the *coordinated method.* This framework implies having a primary focus on the software, but takes some consideration in hardware issues. The hardware issues function as an underlay if it is possible at all to implement the standard system on existing hardware. The practice isn't limited in studying only the standard system it self. It brings out a judgment of whole. Meaning, that it pays regard to the supplier and issues concerning the supplier. This recommended practice could, by that means, be said to be an *overall method.*

Functional Relations

Qualities and questions supporting the understandability of people's comprehensibility in a system.

Qualities ***Questions***

Functionality: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Does the basic function of the software meet the acquirer's needs?
Are its overall capabilities consistent with the requirements of the acquirer's application?
Can the software be run under the acquirer's operating system?

Accessibility: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Was the software available for actual use when it was needed?
Can another user use prevent you from using the system?

Efficiency: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Is the performance adequate for the acquirer's needs?
Are believable performance figures available?
How many users can be on the system before it begins to slow down? What verifiable evidence is available showing that the supplier has tested performance issues in a suitable environment?

Flexibility: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Are the software's input, output, and processing capabilities flexible enough to accommodate the changing requirements of the acquirer's business?
Can the software be adapted to new application?

Stability: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Does the product have a clean, modular design?
Has it been in actual use long enough to make sure that most of its bugs have been cleaned up?
Are there any errors that a user can make that will bring the system down?
What are the recovery capabilities?

Economy: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Is the acquirer's service agreement cost-effective?
In what areas have you found the system to be most cost-effective?
In what areas have you found the system to be least cost-effective?
This practice also includes issues regarding the cost for acquiring and using the software, and what is included in this costs.

Structural Relations

Qualities and questions on issues supporting the understandability and awareness of a system.

Qualities ***Questions***

Rights and Responsibility: The IEEE RPSA embrace this quality by the criteria formulated as question: Are user and file security levels adequate?

Security: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Can unauthorized transactions or programs be run?
Are accounting audit controls satisfactory?
Do accounting audit controls satisfy the acquirer's accountant?

Infological Relations

Qualities and questions supporting the understandability on how meaningfull people experience the system to be.

Qualities ***Questions***

Intelligibility: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Will the software be easy to use?
Is it designed for straightforward operation with a well-documented operating procedure?
Are the reports and screen displays it produces readable, informative, and easy to interpret?
Are help screens provided?

Relevance: The IEEE RPSA embrace this quality by criteria's formulated as questions:
Does the basic function of the software meet the acquirer's needs?
Are its overall capabilities consistent with the requirements of the acquirer's application? (see Functionality)

Knowledge: The IEEE RPSA doesn't embrace this quality.

Competence: The IEEE RPSA embrace this quality by criteria's formulated as questions:
What is the level of technical knowledge required to use and maintain the system?

Motivation: The IEEE RPSA embrace this quality by the criteria formulated as question:
Will the users be enthusiastic about this product?

Social-Technical Approach to COTS Evaluation (STACE)

The Social-Technical Approach to COTS Evaluation (STACE) framework is developed to facilitate a simple, quick and inexpensive social-technical approach to COTS selection process (Kunda, 1999). COTS products are not likely to be modified for a specific customer and can therefore be classified within *completely standardized systems*. COTS software component selection is a process of determining "fitness for use" of previously developed components that are being applied in a new system context. Component selection is also a process for selecting components when a marketplace of competing products exists.

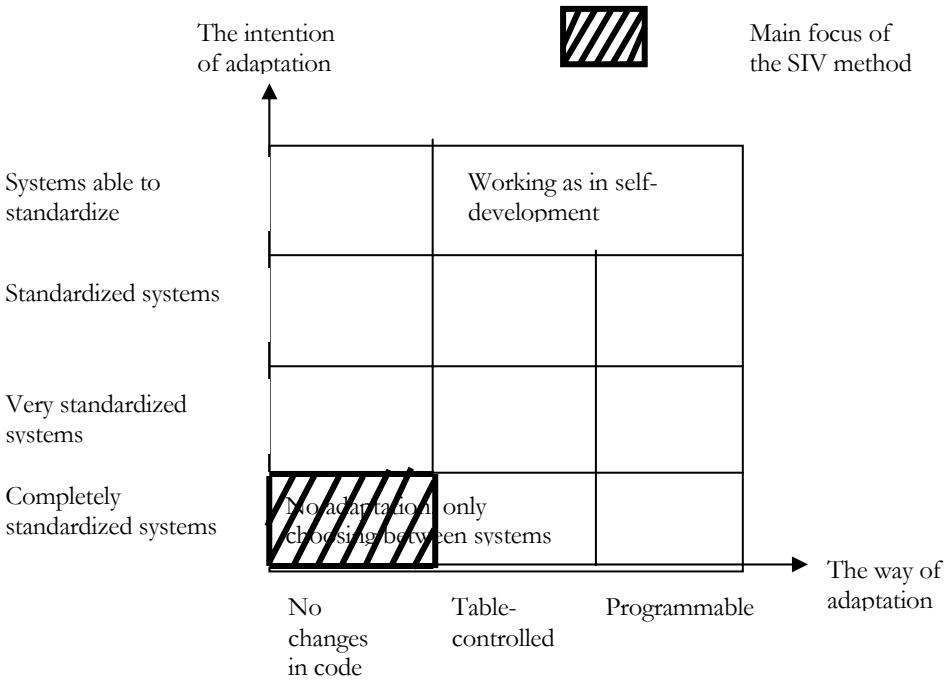

Figure 3.9 Main competence area of the STACE framework.

The STACE framework is limited to a specific area of application, COTS, and can by that means be said to be a *specific method*.

The STACE framework is a complete method for the whole process of evaluating *completely standardized systems*. The method consists of extensive descriptions on the whole process of evaluation from the beginning to the end including both working-steps and templates for documents that needs to be written. The STACE framework can, by that means, be said to be a *maximal method*.

Regarding STACE and how it relates to software and hardware issues the framework follows the *coordinated method*. This framework implies having a primary focus on the software, but takes some consideration in hardware issues. The hardware issues function as an underlay if it is possible at all to implement the standard system on existing hardware. The STACE framework isn't limited in studying only the software it self. It brings out a judgment of whole. Meaning, that it pays regard to the supplier and issues concerning the supplier. This framework can, by that means, be said to be an *overall method*.

Functional Relations

Qualities and questions supporting the understandability of people's comprehensibility in a system.

Qualities ***Questions***

Functionality: The STACE framework embraces compliance issues. Functionality requirements are put together regarding to Customer/Organisations standards, Functionality (domain specific), Organisational policies and Suitability etc, and are weighted according to relevance.

Accessibility: The STACE framework embraces criteria's regarding functional accessibility such as Usability, Responsiveness, Efficiency/Resource etc.

Efficiency: Criteria factors such as functional Effectiveness, Responsiveness are included.

Flexibility: Criteria factors regarding adaptability such as Replaceability, Scalability, Interoperability, etc. are included.

Stability: Criteria factors regarding maintainability such as Understability and factors regarding marketplace variables, such as Product/technology reputation (maturity, stability) are included.

Economy: Criteria's such as Contractual issues, Cost of adapting and integrating, Costs in general, Escrow or buy rights, Licensing arrangements, Product costs, Cost of operation, technology costs, Allocation of resources, Cost justification, embrace this quality.

Structural Relations

Qualities and questions on issues supporting the understandability and awareness of a system.

Qualities ***Questions***

Rights and Responsibility: The STACE framework doesn't include issues regarding rights and responsibility within the system.

Security: Security issues are included as criteria factors.

Infological Relations

Qualities and questions supporting the understandability on how meaningfull people experience the system to be.

Qualities ***Questions***

Intelligibility: Criteria's such as Suitability, Correctness, Usability, Interface issues etc, regarding this quality are included.

Relevance: The STACE framework embraces this quality by including compliance issues such as Suitability, Correctness.

Knowledge: The STACE framework embraces this quality by issues such as Availability of documentation, Availability of training and support, etc.

Competence: Not supported within the STACE framework.

Motivation: Criteria's such as People's attitude embrace this quality.

3.4.4 Principles guiding the Design of Methodological Support

Management as the art of science of improvement can be defined in terms of comprehensibility, understandability (awareness) and meaningsfullness (Churchman, 1978; Checkland, 1989; Hedberg, 1980; Magoulas and Pessi, 1980; Enqusit and Makrygiannis, 1998).

The designer needs to create comprehensibility (overview) to secure awareness and meaningfulness for the benefit of the designer. This view have been verified by the findings of the DELTA project that have accordingly identified and organized the significant needs for successful management in terms of comprehensibility, understandability (awareness) and meaningfulness of the development of the enterprise and its information systems (Enquist et al., 2001).

The need for *comprehensibility* can be explained in terms of the different forces that increase complexity of the development, as well as lack of instruments to manage the complexity and these forces. The lack of comprehensibility results in incompatible and mismatching enterprise and information systems designs e.g. legacy systems that make the development process incomprehensible. (Ibid)

The need for *understandability* can be explained by the bounded rationality of individuals, which is incapable of absorbing uncertainty introduced by different sources (i.e. spatial, temporal, individual, cultural, organizational, institutional, infological, etc.), as well as diversified images of an ever-changing reality. The lack of understandability results in unshared understandability. The rate of change is independent and uncoordinated in several areas fundamental to the enterprise (i.e. knowledge and skills, technology, business concept, information, culture, methodologies and paradigms for development, etc.). Therefore management must coordinate development with the different watches of change and focus timing of coordination. (Ibid)

The need for *meaningfulness* can be justified in terms of goals for improvements. These goals are expected to be specified in both hard and soft measures, but in a dynamic and heterogeneous enterprise environment it is difficult to get a complete and coordinated goal structure. Fact is that management must still act, even with incomplete but otherwise workable and accepted development goals. The lack of meaningfulness results in lack of motivation and commitment. (Ibid)

The DELTA project framework for coordinated development of enterprise and information systems takes its origin from the concepts of comprehensibility, understandability (awareness) and meaningfulness.

Enquist et al. (2001) defines the concept of comprehensibility as "*dealing with the complexity in the reality of the enterprise and designing for comprehensibility*".

The concept of understandability (awareness) is defined as "*creating a coordinated image of the enterprise and the development among stakeholders*". (Ibid)

Finally, the definition of meaningfulness is expressed as "*the reasons for the change/improvements and the coordination of will*" (Ibid).

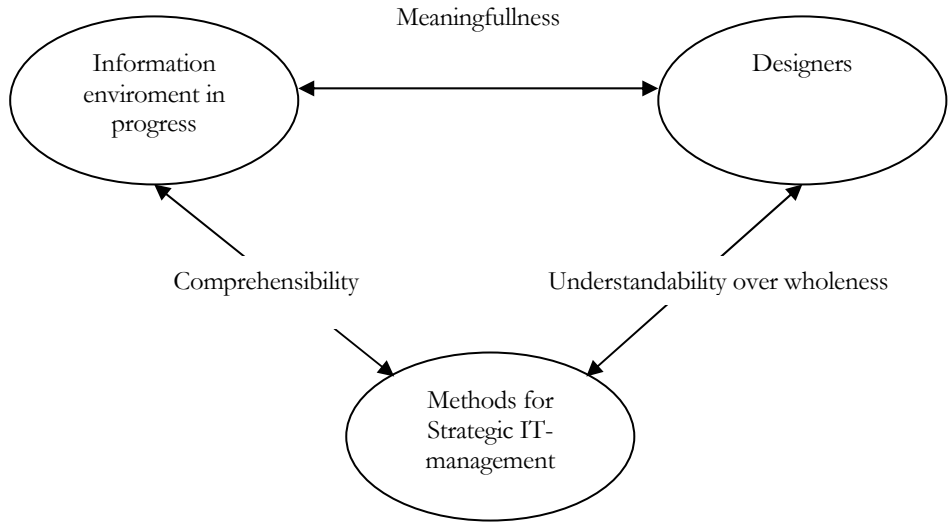

Figure 3.10 Development environment from a methodological point of view.

We have above described an explicit foundation for method integration, consisting of three dimensions. Comprehensibility and understandability are knowledge dependent and create opportunity for meaningfulness. These three dimensions are co-operating and are thereby contributing for the discovery and acceptance of the wholeness.

We have found a number of areas that our developed meta-architecture should clarify. By our means, the goodness of a meta-architectural support can be judged from the below mentioned principles. These principles should be of guidance when choosing/developing meta-architectural support at any given situation. Each principle is finally referred to one of the fundamental requisites: comprehensibility, understandability (awareness), meaningfulness and finally manageability. The principles have been derived inductively through a literature survey of different frameworks.

The basic blocks of an enterprise are people, tasks/purpose, organization and technology. Putting these building blocks together constitute an integrated whole. This implies that development of this whole must include co-ordinated development of all building blocks since they are interdependent. (Ibid)

The basic building blocks of development are according to Enquist et al. (2001):

- Strategic enterprise images
- Stakeholders
- Development goals
- Development process

Strategic Enterprise Images

The enterprise image describes the enterprise identity in its environment. Every organization and every organizational unit has its own set of enterprise images. Enterprise images are based on structures of entities in the enterprise and its environment. When looking at enterprise images there are many perspectives to look from. For instance; an organizational perspective focuses on structures on i.e. organizational hierarchy while a resource perspective sets its focus on buildings, humans, knowledge, information systems and so on. The enterprise is an integrated whole with complex relations between entities described in different perspective.

Stakeholders

The stakeholders are those who have or should have any influence and those that are involved or affected of the organization in any way. It is common to categorize the stakeholders into internal or external depending on their relationship with the organization. The stakeholders can also be categorized depending on where in the organizational hierarchy they are. The different categories have different goals and objectives and this further complicates the development process. (Ibid)

Development Goals

The development goals are defined from the differences between the present and the future enterprise images from which goal structures and goal architectures with the purpose of improvement are created. The development goals are also based on the wants from the different categories of stakeholders, both functional and social goals. Throughout the hierarchy different groups of stakeholders may have various level of knowledge about the mutual goals and therefore the development goals preferably must contain guidelines and measurable goals regarding the changes that the organization is planning to go through. (Ibid)

Development Process

The development process is the process where entities are developed or changed and integrated in the organization. This process changes the organization from one state to another. The development process can consist of activities in formal projects, less explicit group-tasks and even daily local improvement. Some of these activities are controlled by the internal organization while external actors control some. The development processes are usually not that well specified as other processes within the organization that can be both positive and negative for the organization. On the positive side this gives freedom to set up

unique organizational structures for development work. The negative is that it can be difficult to co-ordinate work throughout the organization because nobody has the whole picture of the organization. (Ibid)

Principles related to Enterprise Images

The first group of principles are regarding the choice of instruments that are applicable to the design of enterprise and information systems. (Enquist et al. 2001)

Table 3.1 Principles related to enterprise images

Principle and sources	Interpretation	Management Requisite
Principle of innovativeness	The meta-architectural support should suggest new possibilities of form The meta-architectural support should enable people to see a total range of alternative strategies/solutions	Comprehensibility Meaningfulness
Principle of wholeness	The meta-architectural support should address the enterprise as a whole	Comprehensibility
Principle of architectural goodness	The meta-architectural support should deal directly with architectural forms and their qualities	Comprehensibility
Principles of realism	The meta-architectural l support should lead eventually to structures or processes that can be implemented in IS or in human organizations and job	Meaningfulness
Principle of relativeness	The meta-architectural support should provide concepts that have meaning only in relation to the enterprise and in the context of all concerned architectural dimensions	Meaningfulness Understandability

Principles related to Stakeholders

The second group of principles are regarding the adequacy of particular theories and methodologies to the issues of stakeholders' identifications and involvement. (Ibid)

Table 3.2 Principles related to stakeholders

Principle and sources	Interpretation	Management Requisite
Principle of contradiction	The methodological support should deal with plural and conflicting interests	Meaningfulness
Principle of completeness	The methodological support should take into account the absent and future clients	Meaningfulness
Principle of simplicity	The methodological support should secure the understanding of reality	Understandability
Principle of communicability	The methodological support should support the understanding of complex concepts and communicate them precisely with few, non-technical words	Understandability

Principles related to Development Goals (purpose)

The third group of principle are regarding the designed meta-architecture's capacity to define instruments that promote the integration between thought and action. The primary issue here is the sound coupling between strategy (i.e. though) and projects (i.e. action). Strategy and projects have a common denominator, that is, the ends and means of development. The consequence of this requisite is the selective choice of those instruments that are capable to satisfy this critical issue of development performance. (Ibid)

Table 3.3 Principles related to development purpose/goals

Principle and sources	Interpretation	Management Requisite
Principle of "Here & Now"	The methodological support lead to measurable contribution to the strategic advantages	Meaningfulness
Principle of usefulness	The methodological support should lead to measurable contribution to the strategic advantage	Meaningfulness

Principles related to Development Processes

The last group of principles are regarding the instruments that refer to the most crucial aspects of change and development through projects. Projects are the units of organized and coordinated actions, which presuppose their sound integration to strategy. The process in general, and development in particular, denotes the continuous changes in the states of enterprise. However, not all changes are always desirable. Instead of improvements and harmony, changes can lead to distortions and conflicts. Understanding this fact, one must design the situational meta-architectural support with respect to the instruments that promotes coordination of projects, as well as evaluation of both processes and states of development with respect to the strategy. (Ibid)

Table 3.4 Principles related to development process

Principle and sources	Interpretation	Management Requisite
Principle of usability (K. Lynch)	The methodological support should be used in rapid, partial decisions, with imperfect information, by lay persons who are the direct users of the "place" in question	Manageability
Principle of methodological quality (K. Lynch)	The methodological support should be sufficiently simple, flexible and divisible	Manageability
Principle of temporal significance (K. Lynch)	The methodological support should connect values of very general and long-range importance to that form, and to immediate, practical actions about it	Manageability

4 A Meta-Architecture for Supporting Standard Systems Evaluation

Many methods have appeared in the literature during the last years, each with its own special features, advantages, and unfortunately limitations. Each method was developed with an application area in mind, and therefore they are limited in the scope of applicability. Current approaches and methods for software evaluation do not adequately deal with these human, social, and organizational issues. In this chapter a meta-architecture for standard systems evaluation, which embraces involvement of users in the software evaluation process, will be presented.

4.1 The Design Product

We developed the Meta-Architecture for Standard Systems Evaluation (MASSE) to facilitate a repeatable, systematic, sound, simple, quick and inexpensive approach to standard systems evaluation. The meta-architecture shall embrace the advantages of several existing software evaluation methods and in particular their infological, structural, functional, procedural and substantive issues, in one consistent meta-architecture (see figure 4.1).

MASSE was developed to consolidate some of the most commonly known practices we have been able to identify for standard systems selection.

The MASSE evaluation criteria's are based upon the following existing software evaluation methods:

- IEEE Recommended Practice for Software Acquisition (IEEE RPSA)
- Standardsystem I Verksamheter (SIV)
- Social-Technical Approach to COTS Evaluation (STACE)

Several features separate MASSE from similar methods (see figure 4.2). The main feature is a suggested set of criteria that expands the understanding and evaluation of characteristics beyond commonly evaluated characteristics such as function or cost. Categories of criteria include infological, functional and structural issues, which contributes to increased apprehension of the quality in the information system, and substantive and procedural issues, which contributes to increased security in the development work. Further on, quality in an information system and security in the development work contributes to an accepted and successful implementation outcome.

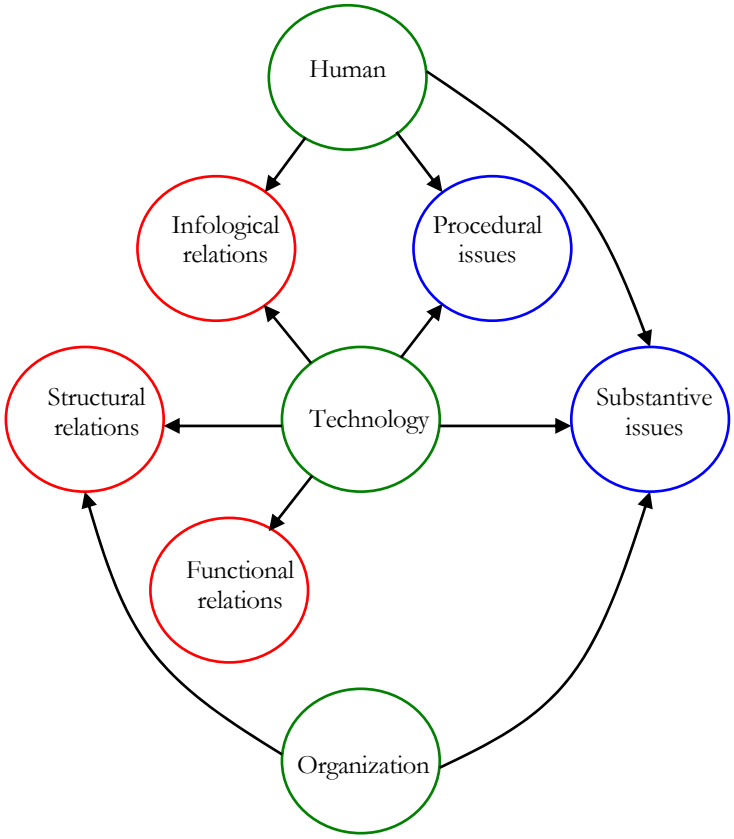

Figure 4.1 Building blocks of the MASSE.

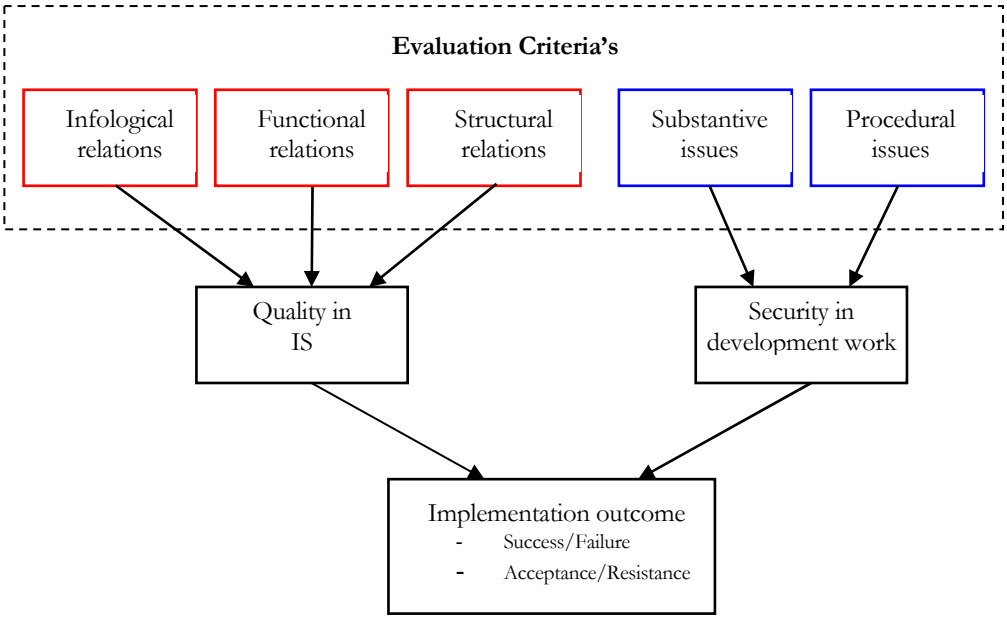

Figure 4.2 The MASSE and IS implementation.

4.2 Evaluation Criteria's

Functional Relations Qualities and questions support the understanding of people's comprehensibility in a system.	
Qualities	**Questions**
Functionality	Functionality requirements are put together regarding to Customer/Organisations standards, Functionality (domain specific), Organisational policies and Suitability etc, and are weighted according to relevance (see 4.3 functionality weighting).
Accessibility	How usable are the functions within the system? How responsive are the functions within the system?
Efficiency	Is the performance adequate for the acquirer's needs? Are believable performance figures available? How many users can be on the system before it begins to slow down? What verifiable evidence is available showing that the supplier has tested performance issues in a suitable environment?
Flexibility	Are the software's input, output, and processing capabilities flexible enough to accommodate the changing requirements of the acquirer's business? Can the software be adapted to new application?
Stability	Does the product have a clean, modular design? Has it been in actual use long enough to make sure that most of its bugs have been cleaned up? Are there any errors that a user can make that will bring the system down? What are the recovery capabilities?
Economy	Is the acquirer's service agreement cost-effective? In what areas have you found the system to be most cost-effective? In what areas have you found the system to be least cost-effective? What was the total cost of acquiring and using the

	software product? Are direct costs included for the price of the software? Are direct costs included for the price of the documentation? What is included in the indirect costs? - Modifying the software - Training personnel - Converting files - Installing the software - Checking out the software - Operating the software - Maintaining the software after installation - Travel expenses

Structural Relations

Qualities and questions on issues that supports the understandability and awareness of a system.

Qualities	Questions
Rights and responsibility	Are user and file security levels adequate?
Security	Can unauthorized transactions or programs be run? Are accounting audit controls satisfactory? Do accounting audit controls satisfy the acquirer's accountant?

Infological Relations

Qualities and questions supporting the understandability on how meaningful people experience the system to be.

Qualities	Questions
Intelligibility	What language is used in the menus and user manuals? Is the system structure and graphical layouts, suited for the users? Will the software be easy to use? Is it designed for straightforward operation with a well-documented operating procedure? Are the reports and screen displays it produces

	readable, informative, and easy to interpret? Are help screens provided?
Relevance	How usable are the functions within the system? How redundant are the functions within the system?
Knowledge	How available is documentation on the system? How available is user training for the system? How available is support?
Competence	What is the level of technical knowledge required to use and maintain the system?
Motivation	Will the users be enthusiastic about this product?

Substantive Issues
In a given situation substantive questions expresses the issue of managing quality with respect to the essential elements in an information environment like IT, people, processes and structures, business environments, and the mutual or inhered inter-dependencies that link these elements together. They concern for instance informational aspects between information systems i.e. systems coordination and systems cooperation. (Bergenstjerna, 2002)

Procedural Issues
In a given situation procedural questions expresses the management of change i.e. the continuing development and management of quality, change decisions and change processes. They are for instance questions of information technology use, of enterprise-wide development, of systems development, of knowledge development as well as the rate of change. (Bergenstjerna, 2002)

4.3 Functionality Weighting

When evaluating functionalities, SIV uses a matrix for weighted comparisons. When using the matrix you fill out the criteria to be evaluated along with its weight. The weight 1-3 reflects the importance of the criteria, 3 is very important and 1 is less important. The points 0-3 reflects the degree of the system fulfilment of this criteria. Weight x Points creates an index on how well the system functions in accordance to the value of the criteria. At the bottom of the matrix you will receive an index telling you of the overall functionality versus criteria-value fulfilment. (Andersen, 1994)

Criterias to be evaluated	**Weight (1-3)**	**Points (0-3)**			**Weight x Points (0-9)**			**Comments**
		Alfa	Beta		Alfa	Beta		
Criteria 1								
Criteria 2								
Criteria 3								
Criteria 4								
And so on …								
Sum:								

5 Applying the Meta-Architecture at OSS Sales

In this chapter we are going to empirically test our meta-architecture, MASSE, in a real case scenario at OSS Sales, Ericsson AB. First we are going to give a brief description of Ericsson and OSS Sales and then move on to describing our situation in this scenario. Then we will account for the substantial issues of the OSS Sales and then perform the evaluation according to our meta-architecture. However, in this evaluation MASSE is not used ideally since we are using an beforehand made requirement specification that we have made earlier prior to when we started writing this thesis. The requirement specification is made according to Ericsson standards and does not imply procedural issues nor does the substantive issues imply the customer needs and wants.

If you are interested in further readings about OSS Sales we recommend reading Appendix B.

5.1 Ericsson

Ericsson is the largest mobile systems supplier in the world. The world's 10 largest mobile operators are Ericsson customers and some 40% of all mobile calls are made through Ericsson systems.

Ericsson provides total solutions covering everything from systems and applications to services and core technology for mobile handsets. With Sony Ericsson complete mobile multi-media products are provided.

Ericsson has been active worldwide since 1876 and has today roughly 82,000 employees in more than 140 countries. The headquarters is located in Stockholm, Sweden. Kurt Hellström is the CEO[1] and President of Ericsson. (Ericsson, 2002)

5.1.1 Operations Support Systems Sales (OSS Sales)

Telecom management within Business Unit Mobile System provides a distributed second line sales support for the mobile market worldwide. The operation manage a perpetual stream of sales support requests from local Ericsson companies around the world. OSS Sales Support covers Commercial Sales support and Technical Sales support. The purpose is to proactively plan and execute market activities, commercial and technical sales support of the product portfolio including all optional functions and services. OSS Sales Managers (commercial responsibility) and Solution Managers (technical responsibility) maintain this task together, equally important. The purpose is to optimise and secure market share in terms of new systems & upgrades, optional products and services, by controlling the sales process and to drive sales activities. How to focus and what functionality to promote is based on business decisions and customer needs.

[1] Chief Executive Officer

The OSS Sales Helpdesk has been created to make a single interface to handle OSS requests that are related to a business opportunity. The new organisation put new responsibilities to the OSS Sales Helpdesk. The product areas to be handled are: GSM-OSS[2], CN-OSS [3]and RANOS[4].

The OSS Sales Managers and Solution Managers are using proven sales methods like "Account Tracker" (Holden based). The OSS Sales Manager has a commercial responsibility, related to a specific number of important GSM customers or a defined BMOS[5] Business & Sales Management unit. The Solution Manager has a product technical responsibility to be carried out both to identified customers and BMOS Business and Sales Management units. The solution Manager will hold a very important responsibility to implement a well-defined contact network between relevant PU[6] organizations as CN and RANOS. In addition to this the Solution Manager will have a responsibility to take part of the development of the 3G[7] OSS "story"/launch package and to carry out this to the market together with the OSS Sales Managers. In some cases OSS Sales Managers and or Solution Mangers will operate locally which means they will have a responsibility based on local prerequisites.

5.2 Our Design Situation

We are going to practice our meta-architecture on OSS Sales. Unfortunately for this thesis the OSS Sales situation only imply creating an underlay for decision of change. We are therefore mostly going to focus on an evaluation among existing standard systems within Ericsson suitable for OSS Sales. Therefore the procedural issues are limited to only management of decision of change and the evaluation focuses on the substantial issues.

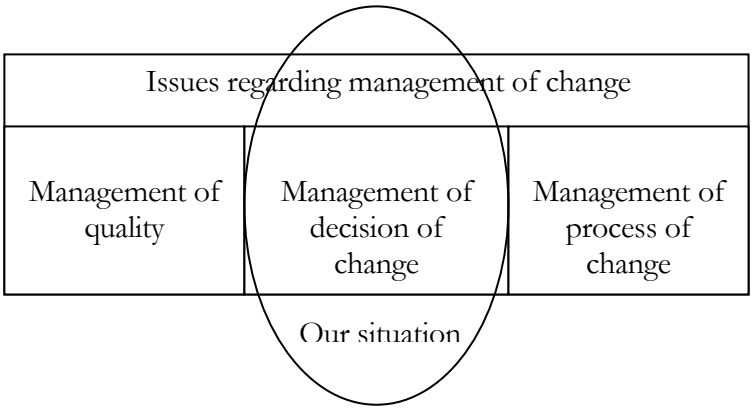

Figure 5.1 Our procedural design situation

[2] Global System for Mobile Communication – Operations Support System

[3] Core Network – Operations Support Systems

[4] Radio Access Network Operation Support

[5] Business Unit Mobile Systems

[6] Product Unit

[7] Third Generation (Mobile communication)

5.3 Substantive Issues

We divided the substantive needs on the new system into functional requirements (input, usability, system and output) and technical requirements. These requirement heritages from the requirement specification we made prior to this thesis. We then interviewed all employees of the OSS Sales to find out what they wanted and needed in the request handling system. After the interviews we gathered all the answers and analysed them together with our supervisor at Ericsson. From out of this we were able to make priorities of every requirement. These priorities are shown in further below during the evaluation of the substantive issues.

5.3.1 Functional Requirements

Input

As input OSS Sales wants the requester to post their requests on the web. This is because they believe the web can save them a lot of "unnecessary" work by forcing the customer/requester to fill out certain fields. Today OSS Sales get their requests through ordinary e-mails and these emails need to be formatted before handled. If the system can do this automatically for them they believe it could save a lot of time. The requests that are coming in often contain attachments and even the outgoing solutions often contain attachments. It is therefore crucial for the system to be able to handle attachments. Furthermore it would be beneficial if the system supports requests to come in via e-mail as well as through web.

The key input functional needs are therefore as follows:

- Web-portal
- Ability to attach files to request and solution
- Ability to send requests via e-mail

Usability

There are a lot of usability functions that OSS Sales want to be able to do with the new system. They believe that they would benefit if working towards a web-GUI [8]since they are often travelling and wants to have access to their personal working area then as well. The real substantive question in this case is as we as the designers see it more like they want to be able to work with their cases regardless of where in the world they are currently located.

The system shall also offer a certain amount of flexibility to be able to meet future demands and changes. Furthermore it is important for the system to be able to log activities so that statistics of various kinds can be presented.

The dispatcher who is delegating the different requests to different handlers wants to be able to see what the handler, that the dispatcher has in mind, currently are working with so that the handler is not drowning with requests. The dispatcher also needs to have the authority to reclaim the rights of a request so that a request not gets timed out because of the handler being home sick or off duty. If a handler doesn't want to take a task (request) he should be able to write a decline of task comment.

[8] Graphical User Interface

When solving requests it is sometimes a good idea to look at previously solved requests to see if the same request has been solved before. It is therefore of great importance to have a knowledge databank of old requests and it is important for this knowledge databank to be structured so that searches can be carried out smoothly. It would also be helpful if the system contained a list of expert to contact when needed. To help finding stored solutions easier it would be nice if there were a function that allowed preview on solutions.

When working with a request it would be nice if the handler had the ability to write a working log to help remember. Furthermore the system shall show improvement compared to the old system. The system must be at least as easy as the prior (Outlook 2000) system, as an example it would be nice if customer fields were filled out automatically. Furthermore it would be nice if the system allowed the requester to view the current status on their request on the web, so that the handler doesn't have to answer a lot of unnecessary questions regarding request statuses. The request status could be followed by the name of the handler if the handler wanted so. When a requester post a request an automatic receipt should be sent when the request is delegated to a handler, it would be nice if one could view these receipts before they are sent to be able to make sure they are correct for just that requester.

The key usability functions are therefore as follows:

- Possibility to work towards a web GUI
- Possibility to modify GUI and functionality
- Ability to log activities
- Ability to show statistics
- Possibility for the dispatcher to view handler status
- Possibility to view dispatcher schedule
- Ability for the dispatcher to reclaim the rights of a task
- Easy to find, sort and view old cases
- Preview list on tasks
- Possibility to write decline of task
- Predefined names on experts to help handler
- Ability to write working log
- Ability for the requester to view the request status on the web
- Option for the handler to show his/her name
- Option to view receipts before they are sent

System

Today every request gets a CaseID. The CaseID´s are currently created manually and due to this it is common that the CaseID being inconsistent regarding both shape and uniqueness. To avoid this problem in the future it would be the best if the system created these CaseID´s itself. The CaseID shall also be ReadOnly to avoid problems like mistyping; this has been a reoccurring problem when using Outlook2000. Furthermore it is beneficial if the system has interface to a personal address database.

Sometimes a requester sends additional information to their requests. It would be beneficial if the system could handle this information automatically.

Requests often have a due-date when the request has to be done. It is of great importance to not forget a request and therefore an alarm system is important so that a request not gets overdue out of sheer forgetfulness. It would also be nice if the system supported holiday control so that requests pass their due date due to holidays.

OSS Sales needs to have a knowledge database consisting of old cases. There is also a need to be able to delimit searches into different areas or categories and therefore it would be beneficial if OSS Sales could structure their own database. When closing a case (request) the handler must write a comment on why.

Since there is a lot of mail corresponding involved in the process of solving a request the system should be able to support both incoming and outgoing email. It would also be nice if the handlers could use the system offline and if the support unit could share their knowledge database with other support units.

The key system functions are therefore as follows:
- Automatically generated CaseID.
- ReadOnly CaseID
- Integration with personal address database
- Automatically updated cases if new information is acquired
- Alarm if case appears to be forgotten
- Own database indexation
- In-Outgoing email support
- Public holiday control
- Not be able to close a case without commenting
- Possibility to use the system offline
- Shared knowledge database with other support units

Output

When a requester posts a request the system should automatically send a receipt back to the requester. Also when done with a request the system shall automatically create an answer out of a template to which the handler can edit if not fully correct.

The key output functions are therefore as follows:
- Automatically generated receipts
- Automatically generated answers from templates.

5.3.2 Technical Requirements

The system must be quick and support multiple users without impact on system performance. To fix and prevent software errors there must be a support unit for those kinds of issues. Most users are using MS Windows environment (NT and 2000) and therefore the client software must support this. English is the general language in the Ericsson concern therefore English shall be supported in the request handling system. Software within Ericsson should be cleared in ESOE (Ericsson Standard Office Environment) in order to be maintained properly by system administration. Approving new software in ESOE takes a lot of time and should be avoided.

The key technical requirements are therefore as follows:

- System performance
- Software support
- System and hardware support
- Client platform requirement
- Sever platform requirement
- Language requirement
- ESOE certification

5.4 Presentation of The Request Handling Systems

5.4.1 World Integrated Helpdesk Gold (WIHGold)

World Integrated Helpdesk Gold was first released in May –00 but earlier versions of WIH has been on the market since 1997. WIH is a core application developed in Remedy/ARS and based on the Ericsson Global IT Services User Support Process for managing IT support, such as request handling, within Ericsson. WIH was originally developed for use with own local server. WIHGold on the other hand is a hotel-based client server system that runs on a host server, which is located at Global IT Services, GIS, in Älvsjö, Sweden. (Method & Tools, 2002)

Global Ericsson helpdesks and expert functions can connect to WIHGold without the need of implementing a whole new standalone system just for their division. This makes WIHGold cost-effective due to all administration is focused to Global IT Services who also is responsible for maintenance, system-support and updates. The quality of WIHGold is assured by following the EITS(Ericsson IT Support) User Support Process. Ericsson Global IT Services will assure that all services and support is delivered with high quality. (Method & Tools, 2002)

The WIHGold client supports Windows (98,2000 and NT) and Unix (Solaris HP-UX). The client is an ESOE (Ericsson Standard Operation Environment) certified Remedy client and can be downloaded from the Method & Tools homepage on the Ericsson intranet. (Rönnberg, 2002)

Characteristic features

- Knowledge database
- Remote access possibility
- Customer web access
- File attachments
- Outgoing e-mail support
- Interactive training included

5.4.2 Answers Questions (AQ)

Answers Questions, AQ, is a system used by Answer Sales-related Questions, ASQ. ASQ is a support function within Ericsson providing a single point entry for support issues from various organizations within Ericsson. ASQ also provides support for external customers having an E-Business relation with Ericsson. There is also a possibility to send requests via

Internet making it possible for ordinary people to ask questions directly to an Ericsson helpdesk. (ASQ, 2002)

ASQ uses their own support system, which is a slightly modified version of WIHGold. The main difference is that they have their own server and database and are therefore not dependent on Ericsson Global IT Services, EGIS, regarding structural issues i.e. database indexation. However, EGIS is still responsible for the maintenance and support for the AQ server. The server is likewise to the WIHGold server located in Älvsjö, Sweden. (ASQ, 2002)

As an AQ client you will need to install an ESOE certified Remedy client. AQ supports MS Windows (98,2000 and NT) and Unix (Solaris and HP-UX). (Rönnberg 2002)

Characteristic features

- Knowledge database
- Remote access possibility
- Customer web access
- File attachments
- Outgoing e-mail support

5.4.3 Global Service and Support System (GS3)

In 1998 the idea of GS3 was first established (Halderman, 1998). GPMS[9] and the GS3 project became the GS3 product (GS3 Support Team, 2002b). GS3 is now used by Ericsson support organizations worldwide. It is a customer service management system, designed to link Ericsson technical support organizations into a global support network. The system makes each support organisation more efficient and promotes sharing of information and knowledge between them.
(GS3 Support Team, 2002a)

GS3 is a tool used for creating, tracking and closing Customer Service Requests, CSR. (Fallström, 2001a) GS3 is based on GPMS (Global Problem Management System) but has a dedicated configuration and is set up in a replicated environment. GPMS is a modified version of Clarify ClearSupport (Nordenborg, 1996). It consists of a server and a client side. The actual databases reside on the servers and the client uses client software to access the database. The database is Sybase based and the 3rd party software used to access it is Clarify's Front office. (Fallström, 2001a).

The primary markets for GS3 are the global support providers within Ericsson. There are normally 3 servers installed worldwide, one in the Europe region, one in the Americas and one in Asia-Pacific. All three servers are real time replicated. As a temporary workaround to performance issues, 2 servers are installed in the Americas. (Fallström, 2001a).

Each server is capable of handling approximately 1000 users simultaneously logged on (Zackrisson, 2001). GS3 is owned and marketed by ERA/GV/RP[10], (Magnusson, 2000).

[9] Global Problem Management System

[10] An Ericsson division

Two types of clients exist for GS3, the desktop client and a web client. The desktop client can be considered as the full client including all necessary functionality, while the web client is limited to only include functionality to create and view CSRs created by the user's own organization. There are client versions for both PC and Solaris. (Fallström, 2001a)

The GS3 desktop client can either be installed using ESOE or by installing each component separately. The ESOE client is certified for use on operating systems; NT 4.0, NT 2000, Windows 95 and Windows 98. The non-ESOE client at LMC[11] can be installed using a batch file. The batch file installs all necessary components and configures the software. This batch file works on operating systems, NT 4.0, Windows 98 and Windows 2000. (Fallström, 2001b)

The actual operation and maintenance of the GS3 system infrastructure is not in the hands of the LMC tools support team. The servers are maintained by GIS, Sybase is maintained by GIS database administrators and the modified Clarify client is designed and supported by the GS3 system administrators in EPA. (Fallström, 2001a)

Characteristic features

- Remote access possibility
- E-mail support
- Customer web access
- Logging of activities
- Possibility to modify GUI and functionality
- Supported interfaces to: MSS[12], PRIMUS [13]and CSR Measurement Tool
- One-day training course available

5.4.4 Service Management Systems (SMS)

Service Management Systems, SMS, Project is the new name for the New Solution Project Phase 2. SMS is targeted at automating the CSR handling process from 1st level support through to the PDU and back. It is customer, rather than product focussed and aims at empowering the 1st Level support organisation with processes and support systems. As with the Knowledge Repository Project the 1st Level Support Organisations of the Market units will be targeted as well as 2nd Level.

SMS is not just another tool. It is Ericsson's strategy for providing Global Support into the future, enabling support of both existing technologies and new generation products. For the first time, Ericsson has developed standard Global CSR Handling Processes, using a single system across All 1st & 2nd line organisations, based upon Ericsson's strategic IT platform (SAP), resting on a common global database, integrated with a central knowledge database. There will be no further requirement to operate distributed servers, support multiple systems (in multiple versions), and redefine variant processes and duplicate solutions. The cost benefits to Ericsson of this support alone many times outweigh the

[11] Ericsson research center in Canada

[12] Maintenance Support System

[13] Existing Ericsson knowledge database

costs of SMS deployment and furthermore provide a way of achieving much greater operational efficiency. (Global Services, 2002)

Computers with Windows 98 and NT platforms shall access SMS through a Terminal Server, TS, set up. PC's on Windows 98/NT that are used today in the support organization will be sufficient to access SMS through the TS solution. No hardware upgrades are necessary. Unix users shall access SMS through the TS set up. Computers with ESOE 2000 (Windows 2000) can access SMS either through the Terminal Server arrangement or through a PC client set up. The PC client set up requires client software, e.g. SAP GUI. (Global Services, 2002)

The SAP GUI for Windows can be installed locally, and runs in a separate window. Another solution is to use a Windows Terminal Server that provides access to the GUI. This eliminates the need to perform the installation on each user's PC. The GUI runs in the right window of the Workplace via the terminal server. (SAP, 2000)

The ongoing managed service will be provided by the Ericsson IT organization. SMS will run on a server that is located in Älvsjö, outside Stockholm. (Global Services, 2002)

Characteristic features

- Remote access possibility
- E-mail support
- File attachments
- Logging of activities
- Supported interfaces to: MSS, MHS[14], TR/Tool [15]and PRIMUS
- Training available by Ericsson University
- 'Corporate Solution' therefore Corporate IT pays for licensing and support

5.5 Evaluation of functional, structural and infological relations

Here we present our attempt to evaluate functional, structural and infological relation within the request handling systems with the support of MASSE.

5.5.1 World Integrated Helpdesk Gold (WIHGold)

Functional Relations
Qualities and questions support the understanding of people's comprehensibility in a system.

Qualities	Questions
Functionality	Functionality requirements are put together regarding to Customer/Organisations standards, Functionality (domain specific), Organisational policies and Suitability etc, and are weighted according to relevance. *Answer: See evaluation of substantive issues.*

[14] Message Handling System

[15] Interface towards MHS

Accessibility	How usable are the functions within the system? *Answer: Most of the functions are usable to OSS Sales. (Demonstration by Method & Tools, Älvsjö, 2002)* How responsive are the functions within the system? *Answer: For client access time from Asia and Americas to the system in Sweden terminal server access is about 2-8 seconds for a normal request from the database). With a normal Remedy client, access is very slow from these distance (30-40 seconds sometimes), but from Sweden the access time is 2-8 seconds. (Hellberg, 2002)*
Efficiency	Is the performance adequate for the acquirer's needs? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Are believable performance figures available? *Answer: Yes, for client access time from Asia and America to system in Sweden and of course accessing the system from within Sweden (Hellberg, 2002)* How many users can be on the system before it begins to slow down? *Answer: No defined limit in simultaneous users. (Method & Tools, 2002)* What verifiable evidence is available showing that the supplier has tested performance issues in a suitable environment? *Answer: The system has been implemented in numerous organizations. (Method & Tools, 2002)*
Flexibility	Are the software's input, output, and processing capabilities flexible enough to accommodate the changing requirements of the acquirer's business? *Answer: As a customer you cannot change the system since the WIHGold/AQ service have to be static to be correct. If you want to add functions to the system you have to contact an administrator. (Method & Tools, 2002)* Can the software be adapted to new application? *Answer: Yes. (Olsson, 2002)*
Stability	Does the product have a clean, modular design? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Has it been in actual use long enough to make sure that most of its bugs have been cleaned up? *Answer: Yes, it has been in use since 2000. (Method & Tools, 2002)* Are there any errors that a user can make that will bring the system down? *Answer: No, none known of (Olsson, 2002)* What are the recovery capabilities? *Answer: Back-ups are made continuously at Methods & Tools. (Demonstration by Method & Tools, Älvsjö, 2002)*
Economy	Is the acquirer's service agreement cost-effective? *Answer: Yes, since Method & Tools and GIS is responsible for maintenance and further development.* In what areas have you found the system to be most cost-effective?

	Answer: By using a system that is hotel-based great economical savings can be done. Since you do not have your own server and do not participate in the software and system development you will split these expenses with other users of the system. In what areas have you found the system to be least cost-effective? *Answer: Since the system being a general system for many helpdesk and expert functions all the functions wanted is not included in the system.* What was the total cost of acquiring and using the software product? *Answer: $9275 the first year. The figure is based on 15 users, single team, 15 floating licences and a 7-form web submission enabled. The cost of implementing 15 floating licences is $3 000, 1 single team implementation is $1 500. To this comes a yearly maintenance charge based on 15 floating licences of $1 575.* *WIHGold offers websubmission, to use this service a charge is based on how many forms that is desired, in our case 7. Websubmission with 7 form is charged on a monthly basis of $200 which is a yearly cost of $2 400. The one time set up charge for web submission single team is $800.* Are direct costs included for the price of the software? *Answer: Yes* Are direct costs included for the price of the documentation? *Answer: Yes* What is included in the direct costs? - Modifying the software - *Answer: No* (Method & Tools, 2002) - Training personnel - *Answer: Yes* (Ibid) - Converting files - *Answer: Yes, but depends on what file and format.* (Ibid) - Installing the software - *Answer: No* (Ibid) - Checking out the software - *Answer: No* (Ibid) - Operating the software - *Answer: Yes* (Ibid) - Maintaining the software after installation - *Answer: Yes* (Ibid) - Travel expenses - *Answer: No* (Ibid)

Structural Relations

Qualities and questions on issues that supports the understandability and awareness of a system.

Qualities	Questions
Rights and responsibility	Are user and file security levels adequate? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)*
Security	Can unauthorized transactions or programs be run? *Answer: No, none aware of. (Rönnborg, 2002)* Are accounting audit controls satisfactory? *Answer: Yes.* Do accounting audit controls satisfy the acquirer's accountant? *Answer: Yes.*

Infological Relations Qualities and questions supporting the understandability on how meaningful people experience the system to be.	
Qualities	**Questions**
Intelligibility	What language is used in the menus and user manuals? *Answer: English. ((Demonstration by Method & Tools, Älvsjö, 2002)* Is the system structure and graphical layouts, suited for the users? *Answer: English. (Demonstration by Method & Tools, Älvsjö, 2002)* Will the software be easy to use? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Is it designed for straightforward operation with a well-documented operating procedure? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Are the reports and screen displays it produces readable, informative, and easy to interpret? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Are help screens provided? *Yes. (Demonstration by Method & Tools, Älvsjö, 2002)*
Relevance	How usable are the functions within the system? *Answer: Many of the system functions are useful to OSS Sales. (Demonstration by Method & Tools, Älvsjö, 2002)* How redundant are the functions within the system? *Answer: The system function redundancy is low. (Demonstration by Method & Tools, Älvsjö, 2002)*
Knowledge	How available is documentation on the system? *Answer: User manuals are downloadable at the Method & Tools homepage. (Method & Tools, 2002)* How available is user training for the system? *Answer: Webcourses are available on the Method & Tools homepage. (Method & Tools, 2002)* How available is support? *Answer: Method & Tools do not have a support phone, but support issues can be sent throughout WIHGold or by contacting your local helpdesk. (Method & Tools, 2002)*
Competence	What level of technical knowledge is required to use and maintain the system? *Answer: No technical knowledge is required to use and maintain the system, but user training is needed.*
Motivation	Will the users be enthusiastic about this product? *Answer: The system is in many ways useful to OSS Sales and therefore we believe the users to be enthusiastic about the system. However it is important to create an understanding among the users on why to use the system, which in turn hopefully creates motivation to use the system.*

5.5.2 Answer Question (AQ)

Functional Relations
Qualities and questions support the understanding of people's comprehensibility in a system.

Qualities	Questions
Functionality	Functionality requirements are put together regarding to Customer/Organisations standards, Functionality (domain specific), Organisational policies and Suitability etc, and are weighted according to relevance. *Answer: See evaluation of substantive issues.*
Accessibility	How usable are the functions within the system? *Answer: Most of the functions are usable to OSS Sales. (AQ being much alike WIHGold). (Demonstration by Method & Tools, Älvsjö, 2002)* How responsive are the functions within the system? *Answer: For client access time from Asia and Americas to the system in Sweden terminal server access is about 2-8 seconds for a normal request from the database). With a normal Remedy client, access is very slow from these distance (30-40 seconds sometimes), but from Sweden the access time is 2-8 seconds. (Hellberg, 2002)*
Efficiency	Is the performance adequate for the acquirer's needs? *Answer: Yes.* Are believable performance figures available? *Answer: Yes, for client access time from Asia and America to system in Sweden and of course accessing the system from within Sweden (Hellberg, 2002)* How many users can be on the system before it begins to slow down? *Answer: No defined limit in simultaneous users. (Rönnborg, 2002)* What verifiable evidence is available showing that the supplier has tested performance issues in a suitable environment? *Answer: The system is implemented at ASQ*
Flexibility	Are the software's input, output, and processing capabilities flexible enough to accommodate the changing requirements of the acquirer's business? *Answer: As a customer you cannot change the system since the WIHGold/AQ service have to be static to be correct. If you want to add functions to the system you have to contact an administrator. (Method & Tools, 2002)* Can the software be adapted to new applications? *Answer: Yes. (Olsson, 2002)*
Stability	Does the product have a clean, modular design? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Has it been in actual use long enough to make sure that most of its bugs have been cleaned up?

	Answer: Yes, WIHGold has been in use since 2000 and AQ is built from out of WIHGold. (Method & Tools, 2002) Are there any errors that a user can make that will bring the system down? *Answer: No, none known of. (Demonstration by Method & Tools, Älvsjö, 2002)* What are the recovery capabilities? *Answer: Back-ups are made continuously at GIS. (Hellberg, 2002)*
Economy	Is the acquirer's service agreement cost-effective? *Answer: Yes, since Method & Tools and GIS is responsible for maintenance and further development.* In what areas have you found the system to be most cost-effective? *Answer: By using a system that is hotel-based great economical savings can be done. Since you do not have your own server and do not participate in the software and system development you will split these expenses with other users of the system.* In what areas have you found the system to be least cost-effective? *Answer: Since the system being a general system for many helpdesk and expert functions all the functions wanted is not included in the system.* What was the total cost of acquiring and using the software product? *Answer: $6075 the first year. The figure is based on 15 users, single team, 15 floating licences and a 7-form web submission enabled. The cost of implementing 15 floating licences is $3 000, 1 single team implementation is $1 500. To this comes a yearly maintenance charge based on 15 floating licences of $1 575.* *AQ offer no web access therefore is AQ appearing a bit cheaper than WIHGold.* Are direct costs included for the price of the software? *Answer: Yes* Are direct costs included for the price of the documentation? *Answer: Yes* What is included in the direct costs? - Modifying the software - *Answer:* No (Method & Tools, 2002) - Training personnel - *Answer:* No (Ibid) - Converting files - *Answer:* Yes, but depends on what file and format. (Ibid) - Installing the software - *Answer:* No (Ibid) - Checking out the software - *Answer:* No (Ibid) - Operating the software - *Answer:* Yes (Ibid) - Maintaining the software after installation - *Answer:* Yes (Ibid) - Travel expenses - *Answer:* No (Ibid)

Structural Relations

Qualities and questions on issues that supports the understandability and awareness of a system.

Qualities	Questions
Rights and responsibility	Are user and file security levels adequate? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)*
Security	Can unauthorized transactions or programs be run? *Answer: No, none aware of. (Rönnborg, 2002)* Are accounting audit controls satisfactory? *Answer: Yes.* Do accounting audit controls satisfy the acquirer's accountant? *Answer: Yes.*

Infological Relations

Qualities and questions supporting the understandability on how meaningful people experience the system to be.

Qualities	Questions
Intelligibility	What language is used in the menus and user manuals? *Answer: No specific user manuals for AQ. If you know how to use WIHGold then you know how to use AQ. (Rönnborg, 2002)* Is the system structure and graphical layouts, suited for the users? *Answer: English. (Demonstration by Method & Tools, Älvsjö, 2002)* Will the software be easy to use? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Is it designed for straightforward operation with a well-documented operating procedure? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002*) Are the reports and screen displays it produces readable, informative, and easy to interpret? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)* Are help screens provided? *Answer: Yes. (Demonstration by Method & Tools, Älvsjö, 2002)*
Relevance	How usable are the functions within the system? *Answer: Many of the system functions are useful to OSS Sales. (Demonstration by Method & Tools, Älvsjö, 2002)* How redundant are the functions within the system? *Answer: The system function redundancy is low. (Demonstration by Method & Tools, Älvsjö, 2002)*
Knowledge	How available is documentation on the system? *Answer: User manuals of WIHGold can be downloaded at Method & Tools homepage, the systems are pretty much alike. (Rönnborg, 2002)*

	How available is user training for the system? *Answer: Webcourses are available on the Method & Tools homepage.(Only for WIHGold) (Method & Tools, 2002)* How available is support? *Answer: Method & Tools do not have a support phone, but support issues can be sent throughout ASQ or by contacting your local helpdesk. (Method & Tools, 2002)*
Competence	What level of technical knowledge is required to use and maintain the system? *Answer: No technical knowledge is required to use and maintain the system, but user training is needed.*
Motivation	Will the users be enthusiastic about this product? *Answer: The system is in many ways useful to OSS Sales and therefore we believe the users to be enthusiastic about the system. However it is important to create an understanding among the users on why to use the system, which in turn hopefully creates motivation to use the system.*

5.5.3 Global Service and Support System (GS3)

Functional Relations
Qualities and questions support the understanding of people's comprehensibility in a system.

Qualities	**Questions**
Functionality	Functionality requirements are put together regarding to Customer/Organisations standards, Functionality (domain specific), Organisational policies and Suitability etc, and are weighted according to relevance. *Answer: See evaluation of substantive issues.*
Accessibility	How usable are the functions within the system? *Answer: Most of the functions are usable to OSS Sales. (Fallström, 2001b; Stolpe, 1999; Magnusson, 2000; Kallerman, 1999; Hexeberg-Schoultz, 2002)* How responsive are the functions within the system? *Answer: The client is well designed for performance. Observed performance is good except that users can run unlimited queries, and if these use too much resource then other users' performances are severely impacted. (Dilks, 2002)*
Efficiency	Is the performance adequate for the acquirer's needs? *Answer: Yes, the client is well designed for performance. Actual performance is therefore determined by the database configuration and platform dimensions. The supplier gives guidance for each supported database. Observed performance is good except that users can run unlimited queries, and if these use too much resource then other users' performances are severely impacted. (Dilks, 2002)* Are believable performance figures available? *Answer: Yes, GS3 System Monitor (http://gs3.ericsson.se/gs3_monitor/) monitors system performance, available licenses, free FTP[16] space, and notification of server operational disturbance.)*

[16] File Transfer Protocol

	How many users can be on the system before it begins to slow down? *Answer: There are normally 3 servers installed worldwide (Fallström, 2001). Each server is capable of handling approximately 1000 users logged on (Zackrisson, 2001).* What verifiable evidence is available showing that the supplier has tested performance issues in a suitable environment? *Answer: The system has been implemented in numerous organizations. (Dilks, 2002)*
Flexibility	Are the software's input, output, and processing capabilities flexible enough to accommodate the changing requirements of the acquirer's business? *Answer: As a customer you cannot change the system since the GS3 service have to be static to be correct. If you want to add functions to the system you have to contact an administrator. (Hexeberg-Schoultz, 2002)* Can the software be adapted to new application? *Answer: Other CRM applications are covered. It is not, however, a generic application builder. (Dilks, 2002)*
Stability	Does the product have a clean, modular design? *Answer: Yes. (Demonstration by Hexeberg-Schoultz, 2002)* Has it been in actual use long enough to make sure that most of its bugs have been cleaned up? *Answer: Yes, but the current release, R3 has known bugs. These have been corrected in R4 but this has not been rolled-out due to impending replacement of GS3 by SMS. (Dilks, 2002)* Are there any errors that a user can make that will bring the system down? *Answer: Users can (rarely) crash their client. The system providers has never had a server crash or had to carry out a recovery procedure. (Dilks, 2002)* What are the recovery capabilities? *Answer: The actual operation and maintenance of the GS3 system infrastructure is not in the hands of the LMC tools support team. The servers are maintained by GIS, Sybase is maintained by GIS database administrators and the modified Clarify client is designed and supported by the GS3 system administrators in EPA[17].* *(Fallström, 2001a)*
Economy	Is the acquirer's service agreement cost-effective? *Answer: Yes.* In what areas have you found the system to be most cost-effective? *Answer: By using a system that is hotel-based great economical savings can be done. Since you do not have your own server and do not participate in the software and system development you will split these expenses with other users of the system.* *GS3 is used by Ericsson support organizations worldwide. It is a customer service management system, designed to link Ericsson technical support organizations into a global support network. The system makes each support organisation more efficient and promotes sharing of information and knowledge between them.* *(GS3 Support Team, 2002a)* In what areas have you found the system to be least cost-effective?

[17] Environmental Protection Agency

	Answer: Since the system being a general system for many helpdesk and expert functions all the functions wanted is not included in the system. What was the total cost of acquiring and using the software product? *Answer: The cost for GS3 is included in a total IT support cost that is keyed out to each MU based on reported head count for Customer Services, System Support personnel. Every MU will be invoiced per reported head count, irrelevant of the actual tool usage.* *For 2000, the cost is calculated to be 10.000 SEK per reported support person and year, provided that no major change in strategy is made. GS3 will be shut down at the end of 2002, so thry don't get "new" groups anymore. It isn't worth the effort en money for those few months.* *(in't Groen, 2002)* Are direct costs included for the price of the software? *Answer: Yes. Software to install the GS3 client can be found on the web, so no extra costs for this (in't Groen, 2002).* Are direct costs included for the price of the documentation? *Answer: The documentation can be found on a web page (in't Groen, 2002).* What is included in the direct costs? - Modifying the software *Answer: Yes. All users can request changes but the modification will only be done if seen as part of and beneficial (save cost) for the global organisation (Almroth, 2002).* - Training personnel *Answer: Yes (in't Groen, 2002).* - Converting files Answer: No *(in't Groen, 2002).* - Installing the software *Answer: Yes. The information on how to install the client, and the software to install the client, can be found on the web for free (in't Groen, 2002).* - Checking out the software *Answer: Yes (in't Groen, 2002).* - Operating the software *Answer: Yes (in't Groen, 2002).* - Maintaining the software after installation *Answer: Yes (in't Groen, 2002).* - Travel expenses *Answer: Yes, but only for activities seen as part of operating the system (Almroth, 2002).*

Structural Relations

Qualities and questions on issues that supports the understandability and awareness of a system.

Qualities	Questions
Rights and responsibility	Are user and file security levels adequate? *Answer: Yes (Dilks, 2002).*
Security	Can unauthorized transactions or programs be run? *Answer: No (Dilks, 2002).*

	Are accounting audit controls satisfactory? *Answer: Yes.* Do accounting audit controls satisfy the acquirer's accountant? *Answer: Yes.*

Infological Relations

Qualities and questions supporting the understandability on how meaningful people experience the system to be.

Qualities	Questions
Intelligibility	What language is used in the menus and user manuals? *Answer: English (Demonstration by Hexeberg-Schoultz, 2002).* Is the system structure and graphical layouts, suited for the users? *Answer: Yes.* Will the software be easy to use? *Answer: Yes.* Is it designed for straightforward operation with a well-documented operating procedure? *Answer: Yes (Dilks, 2002).* Are the reports and screen displays it produces readable, informative, and easy to interpret? *Answer: Yes (Dilks, 2002).* Are help screens provided? *There are no help screens provided (Dilks, 2002).*
Relevance	How usable are the functions within the system? *Answer: Most of the system functions are useful to OSS Sales.* How redundant are the functions within the system? *Answer: The system function redundancy is low.*
Knowledge	How available is documentation on the system? *Answer: Documentation is very available and easy accessible at http://gs3.ericsson.se.* How available is user training for the system? *Answer: There is a one-day course for the end users, and one day extra for the Local Data Administrator. There is a trainer coming over to provide the user course.* *(in't Groen, 2002)* How available is support? *Answer: Very available. There is a telephone list for GS3 user & site data and application support at http://gs3.ericsson.se.*
Competence	What level of technical knowledge is required to use and maintain the system? *Answer: No technical knowledge is required to use and maintain the system, but user training is needed.*
Motivation	Will the users be enthusiastic about this product?

	Answer: The system is in many ways useful to OSS Sales and therefore we believe the users to be enthusiastic about the system. However it is important to create an understanding among the users on why to use the system, which in turn hopefully creates motivation to use the system.

5.5.4 Service Management Systems (SMS)

Functional Relations
Qualities and questions support the understanding of people's comprehensibility in a system.

Qualities	Questions
Functionality	Functionality requirements are put together regarding to Customer/Organisations standards, Functionality (domain specific), Organisational policies and Suitability etc, and are weighted according to relevance. *Answer: See evaluation of substantive issues.*
Accessibility	How usable are the functions within the system? *Answer: Most of the functions are usable to OSS Sales. (Global Services, 2002; Willems, Pedersen and Åkerlind, 2002; SMS Functional Team, 2002)* How responsive are the functions within the system? *Answer: The system is well designed for performance since it is built on SAP CRM 3.0 (Global Services, 2002).*
Efficiency	Is the performance adequate for the acquirer's needs? *Answer: Yes, the client is well designed for performance. Actual performance is therefore determined by the database configuration and platform dimensions.* *(Global Services, 2002)* *With SAP CRM as a ground and a well-configured database there should not be any performance problems.* Are believable performance figures available? *Answer: The SMS project is currently in its Pilot phase. Deployment of the system is planned to start on August 30st, 2002. The planned finish date is December 31st, 2002. Therefore, this kind of information is still not available. (Joste, 2002)* How many users can be on the system before it begins to slow down? *Answer: No defined number.* What verifiable evidence is available showing that the supplier has tested performance issues in a suitable environment? *Answer: The SAP CRM has been implemented in numerous organizations and the SMS has been carefully tested before its roll out (Global Services, 2002; Lowegard, 2002).*
Flexibility	Are the software's input, output, and processing capabilities flexible enough to accommodate the changing requirements of the acquirer's business? *Answer: The Functional Requirements defined to date have been defined to meet the Ericsson Global Standard. Therefore local processes should conform to this standard. However, in the unlikely event that there are business-critical requirements that are not met by the Global Standard, there was a Change*

	Request Procedure which was open until February 17th 2002. *However, changes identified past 17th February, which are considered to be of major importance, shall be escalated via management to either Björn Wedén and/or Mats Anvret.* *(Global Services, 2002)* Can the software be adapted to new applications? *Answer: Yes* (Ibid)
Stability	Does the product have a clean, modular design? *Answer: Yes. This is observed by the online demo available at: http://globalservices.ericsson.se/cs/e-supportsystems/smsproject/* Has it been in actual use long enough to make sure that most of its bugs have been cleaned up? *Answer: The SMS project is currently in its Pilot phase meaning that this kind of information is still not available. However, the system is based upon mySAP CRM 3.0 that has been tested and used long enough by numerous organizations.* (Ibid) Are there any errors that a user can make that will bring the system down? *Answer: Users can rarely crash their client that is SAP GUI and has been tested and used in numerous organizations worldwide.* (Ibid) What are the recovery capabilities? *Answer: The ongoing managed service will be provided by the Ericsson IT organization. SMS will run on a server that is located in Älvsjö, outside Stockholm. In case anything happens that causes this server to stop work properly service will be switched over to a different server at Telefonplan.* (Ibid)
Economy	Is the acquirer's service agreement cost-effective? *Answer: Yes.* In what areas have you found the system to be most cost-effective? *Answer: SMS is not just another tool. It is Ericsson's strategy for providing Global Support into the future, enabling support of both existing technologies and new generation products. For the first time, Ericsson has developed standard Global CSR Handling Processes, using a single system across ALL 1st & 2nd line organisations, based upon Ericsson's strategic IT platform (SAP), resting on a common global database, integrated with a central knowledge base. There will be no further requirement to operate distributed servers, support multiple systems (in multiple versions), and redefine variant processes and duplicate solutions. The cost benefits to Ericsson of this support alone many times outweigh the costs of SMS deployment and furthermore provide a way of achieving much greater operational efficiency.* *(Global Services, 2002)* In what areas have you found the system to be least cost-effective? *Answer: Since the system being a general system for many helpdesk and expert functions all the functions wanted is not included in the system.* What was the total cost of acquiring and using the software product? *Answer: There is no numbers available at the moment because the SMS project is still under Pilot Phase. The SMS Project will cover the costs of the Technical and Functional teams which are responsible for system definition and configuration. The SMS Project will pay for training of "SMS Coaches" in each of the regions (approximately 1 trainer to 30 users). The Project also provides Regional Deployment support staff that will hold workshops locally. ISM and GIS will support infrastructure in the usual manner.*

	All other costs are borne by the local and regional Support Units. This will include Local Project Management, Resources required for preparation ahead of deployment, Travel and Accommodation expenses for Training and LPM workshops and any other activity that the Support Unit needs to execute in order to be ready to use SMS. *The Project pays for cost of delivery i.e. external and Ericsson consultants.* *SMS is a 'Corporate Solution' therefore Corporate IT pays for licensing and support.* *(Global Services, 2002)* Are direct costs included for the price of the software? *Answer: Yes. Software to install the client, SAP GUI, can be found on the web, so no extra costs for this.* Are direct costs included for the price of the documentation? *Answer: The documentation can be found on a web page.* What is included in the direct costs? - Modifying the software *Answer: Yes. All users can request changes but the modification will only be done if seen as part of and beneficial (save cost) for the global organisation (Global Services, 2002)..* - Training personnel *Answer: Yes (Global Services, 2002).* - Converting files Answer: No *(Global Services, 2002).* - Installing the software *Answer: Yes. The information on how to install the client, and the software to install the client, can be found on the web for free (Global Services, 2002).* - Checking out the software *Answer: Yes (Global Services, 2002).* - Operating the software *Answer: Yes (Global Services, 2002).* - Maintaining the software after installation *Answer: Yes (Global Services, 2002).* - Travel expenses *Answer: Yes, but only for activities seen as part of operating the system (Global Services, 2002).*

Structural Relations

Qualities and questions on issues that supports the understandability and awareness of a system.

Qualities	**Questions**
Rights and responsibility	Are user and file security levels adequate? *Answer: Yes (Willems, Pedersen and Åkerlind, 2002).*
Security	Can unauthorized transactions or programs be run? *Answer: No (Willems, Pedersen and Åkerlind, 2002).* Are accounting audit controls satisfactory? *Answer: Yes.*

	Do accounting audit controls satisfy the acquirer's accountant? *Answer: Yes.*

Infological Relations Qualities and questions supporting the understandability on how meaningful people experience the system to be.	
Qualities	**Questions**
Intelligibility	What language is used in the menus and user manuals? *Answer: English.* Is the system structure and graphical layouts, suited for the users? *Answer: Yes.* Will the software be easy to use? *Answer: Yes.* Is it designed for straightforward operation with a well-documented operating procedure? *Answer: Yes.* Are the reports and screen displays it produces readable, informative, and easy to interpret? *Answer: Yes.* Are help screens provided? *Answer: Yes.*
Relevance	How usable are the functions within the system? *Answer: Most of the system functions are useful to OSS Sales.* How redundant are the functions within the system? *Answer: The system function redundancy is low.*
Knowledge	How available is documentation on the system? *Answer: Documentation is very available and easy accessible at http://globalservices.ericsson.se/cs/e-supportsystems/smsproject/* How available is user training for the system? *Answer: In addition to the Regional SMS Coach Training, materials will be made available to SMS Coaches to deliver web-based training and the methods and techniques of using this approach. This will minimize the requirement for formal class training and facilitate ongoing training requirements beyond the closure of the SMS Project.(Global Services, 2002)* How available is support? *Answer: The SMS project is currently in its Pilot phase. Deployment of the system is planned to start on August 30st, 2002. The planned finish date is December 31st, 2002. Therefore, this kind of information is still not available. (Joste, 2002)*
Competence	What level of technical knowledge is required to use and maintain the system? *Answer: No technical knowledge is required to use and maintain the system, but user training is needed.*
Motivation	Will the users be enthusiastic about this product? *Answer: The system is in many ways useful to OSS Sales and Ericsson, therefore we believe the users to be*

	enthusiastic about the system. However it is important to create an understanding among the users on why to use the system, which in turn hopefully creates motivation to use the system.

5.6 Evaluation of Substantive Issues

Here we will present our weighted evaluation of the substantive issues. The substantive issues in this case reflect the functional and technical requirements. At the bottom of this matrix you can see the final result of the weighted evaluation.

Critical factors	**Weight (1-3)**	**Points (0-3)**				**Weight x Points (0-9)**				**Comments**
		WIH Gold	AQ	GS3	SMS	WIH Gold	AQ	GS3	SMS	
Ability to send requests via web-portal	3	3	2	3	2	9	6	9	6	AQ: Have to use ASQ web submissions. SMS: Have to use CSR Writer tool.
Ability to send requests via e-mail	2	0	0	3	3	0	0	6	6	
Ability for the requester to attach files to the request	3	0	0	3	3	0	0	9	9	
Improvement in functionality and usability	3	2	2	3	3	6	6	9	9	
Automatically filled out fields (requester information)	2	3	3	3	3	6	6	6	6	
Option to view receipts before they are sent	1	0	0	0	0	0	0	0	0	
Possibility to work towards a web-GUI	3	3	0	0	0	9	0	0	0	
Ability for the requester to check the request-status on the web	2	3	3	3	2	6	6	6	4	SMS: Have to use CSR Writer tool.
Option for the handler to show his/her name	1	0	0	0	0	0	0	0	0	
Possibility to modify GUI and functionality	3	1	1	1	1	3	3	3	3	WIHGold/AQ/GS3/SMS: Very limited possibilities
Ability to log activities	3	3	3	3	3	9	9	9	9	
Ability to show statistics: see ref.	3	3	3	3	3	9	9	9	9	
Possibility to view handler status	3	3	3	3	3	9	9	9	9	

Possibility to view dispatcher schedule	1	0	0	0	0	0	0	0	0	
Ability for the dispatcher to reclaim the rights of a task	3	3	3	3	3	9	9	9	9	
Possibility to write a comment on decline of task	2	3	3	3	3	6	6	6	6	
Preview list on tasks	2	3	3	3	3	6	6	6	6	
Ability to write working log	2	3	3	3	3	6	6	6	6	
Predefined names on experts to help handler	2	0	0	0	3	0	0	0	6	
Easy to find, sort and view old cases	3	3	3	3	3	9	9	9	9	
Automatically generated caseID	3	3	3	3	3	9	9	9	9	
Read-Only caseID	3	3	3	3	3	9	9	9	9	
Integration with LDAP or similar	2	3	3	3	3	6	6	6	6	
Automatically updated cases if new information is acquired	3	0	0	3	3	0	0	9	9	
Alarm if case appears to be forgotten	3	3	3	3	0	9	9	9	0	
Own database indexation	3	0	0	0	0	0	0	0	0	
Not be able to close a case without commenting	1	0	0	3	3	0	0	3	3	
Incoming e-mail support	3	0	0	3	3	0	0	9	9	
Outgoing e-mail support	3	3	3	3	3	9	9	9	9	
Possibility to use the system offline	2	0	0	0	0	0	0	0	0	
Shared knowledge-database with other support-units	1	3	3	3	3	3	3	3	3	
Public holiday control	2	0	0	0	0	0	0	0	0	

Automatically generated receipts	3	3	3	3	3	9	9	9	9	
Automatically generated answers from templates	3	3	3	0	3	9	9	0	9	
Ability to attach files to outgoing mail	3	0	0	3	3	0	0	9	9	
Acceptable system performance	3	3	3	3	3	9	9	9	9	
Software support	3	3	3	3	3	9	9	9	9	
System and hardware support	3	3	3	3	3	9	9	9	9	
Client platform requirement	3	3	3	3	3	9	9	9	9	
Server platform requirement	3	3	3	3	3	9	9	9	9	
Language requirement	3	3	3	3	3	9	9	9	9	
ESOE certification	3	3	3	3	3	9	9	9	9	
Sum:						**228**	**216**	**258**	**259**	

5.7 Summary of the Evaluation using MASSE

The evaluation was delimited to only evaluate the substantive issues of OSS Sales. These issues were taken from a requirement specification made prior to this thesis. This requirement specification was made according to Ericsson standards. The outcome of the evaluation is illustrated in the figures below. A small reservation must be made due to a few questions in the MASSE checklist is not of "Yes" or "No" nature. In those cases we will interpret a positive answer as a "Yes". Furthermore, if most of the criteria's are fulfilled within in an area the whole are is given the status fulfilled. Therefore the figures below are rather to see as a first overview and if there is further interest we recommend reading the full answers in the empirical view chapter.

✓ Fulfilled ❍ Partly fulfilled - Not fulfilled

Functional relations

WIHGold	**228**	❍	✓	❍	✓	✓
AQ	**216**	❍	✓	❍	✓	✓
GS3	**258**	✓	✓	❍	✓	✓
SMS	**259**	✓	✓	❍	✓	✓
	Functionality	Accessibility	Efficiency	Flexibility	Stability	Economy

Structural relations

	Rights & responsibilities	Security
WIHGold	✓	✓
AQ	✓	✓
GS3	✓	✓
SMS	✓	✓

Infological relations

	Intelligibility	Relevance	Knowledge	Competence	Motivation
WIHGold	✓	✓	✓	✓	✓
AQ	✓	✓	✓	✓	✓
GS3	✓	✓	✓	✓	✓
SMS	✓	✓	✓	✓	✓

5.8 Recommendations

According to the evaluation between four request handling systems for OSS Sales we have found two of them being the most advantageous for OSS Sales. The two are GS3 and SMS. Out of the two we have found SMS to be slightly more interesting since it being a very recently developed system and is based on SAP, which in turn contributes to the Ericsson strategic platform. This is very good since multiple and/or duplicate solutions can be avoided helping keeping the organizational costs down. During the evaluation we have also learnt that GS3 will shut down at the end of 2002 in favour of SMS. This leads us to a single recommendation of SMS.

6 Analysis

6.1 The Creation of a Meta-Architecture

Management as the art of science of improvement can be defined in terms of comprehensibility, understandability (awareness) and meaningfulness. The designer needs to create comprehensibility/overview to secure awareness and meaningfulness for the benefit of the designer. Comprehensibility and understandability are dependent on knowledge and creates opportunity for meaningfulness. These three dimensions are co-operating and are thereby contributing for the discovery and acceptance of the wholeness.

From out of this we found the functional, structural and infological aspects, which comprehended with the three basic blocks for acceptance.

The goodness of a meta-architectural support can, by our means, be judged from principles. These principles should, by our means, be of guidance when choosing/designing meta-architectural support at any given situation. Finally, each principle is referred to one of the fundamental requisites: comprehensibility, understandability (awareness), meaningfulness and finally manageability.

We have furthermore chosen three commonly used methods for software evaluation and in particular standard systems evaluation. These three methods has been analysed to find out what knowledge they provide is useful for us. We have used the functional, structural and infological perspectives for the analysis.

To begin the analysis we started out by classifying the methods according to Nilsson's way of classification. This helped us in getting a first overview of what to expect from the method and also create a fundamental knowledge of the methods.

6.2 Classifications of Methods

Both SIV method and IEEE RPSA are maximal methods supporting a general field of competence by that meaning they are supportive of the whole process of standard system acquisitioning. The STACE framework on the other hand has a specific field of competence and that is the evaluation part in the acquisition process of standard systems.

When it comes to how the methods relate to hardware and supplier issues all three supports the coordinated method. The coordinated method takes some consideration in hardware issues but don't let them become too dominant since it could come to affect the evaluation of the system itself too much. Even regarding supplier issues all three methods are supportive of the overall method, which means more or less what the coordinated method means to hardware issues. The overall method aims to find the best combination between system and its supplier.

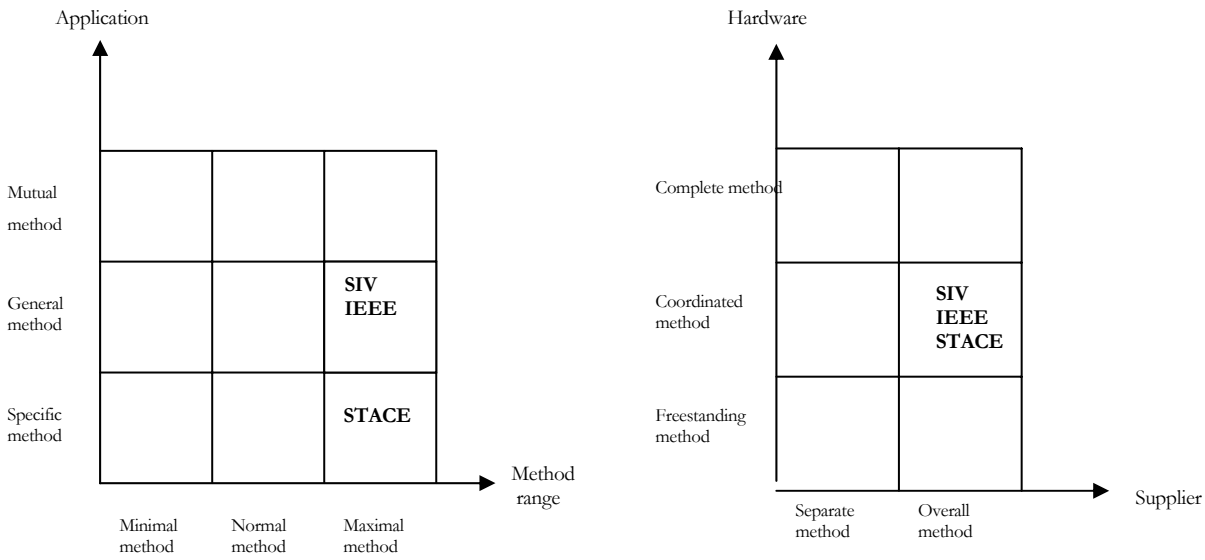

Figure. 6.1 Classification of chosen available methods (SIV, IEEE RPSA and STACE).

6.2 Available Knowledge in Chosen Existing Methods

When analysing the methods above from a functional, structural and infological perspective according to the criteria's of Hage, Kerim and Zamirian (2002) we can clearly see flaws regarding fulfilments.

Functional relations

Qualities **Arguments for integration of methods regarding qualities**

Functionality: The IEEE RPSA has a lack of issues/criteria's regarding this quality. The SIV-method has a shortage of aspects in its criteria's regarding this quality because of its limitation to system functionality requirements that are evaluated by the designer. SIV doesn't specify what issues should be covered within the requirement specification. The STACE framework indicates that Customer/Organizations standards, Functionality (domain specific), Organizational policies and Suitability should be covered within the evaluated requirement specification.

Accessibility: The IEEE RPSA and SIV-method don't pay attention to this quality on the contrary to the STACE framework.

Efficiency: This quality is covered by all three methods but the IEEE RPSA has the right formulation for the issues regarding this quality that embraces the criteria's brought up by the two other methods.

Flexibility: This quality is covered by the SIV-method and the IEEE RPSA. But the IEEE RPSA has the right formulation for the issues regarding this quality that embraces the criteria's brought up by the SIV-method. The STACE framework has a lack of issues/criteria's regarding this quality.

Stability: The SIV-method has a lack of issues/criteria's regarding this quality. The STACE framework method has a shortage of, and a different slant on, aspects in its criteria's regarding this quality. The IEEE RPSA on the other hand, embraces this quality correctly and has the right formulation for the issues regarding this quality.

Economy: This quality is covered by all three methods but the IEEE RPSA has the right formulation for the issues regarding this quality that embraces the criteria's brought up by the two other methods.

Structural relations

Qualities **Arguments for integration of methods regarding qualities**

Rights and Responsibility: The SIV-method and the STACE framework have a lack/shortage of issues/criteria's regarding this quality. The IEEE RPSA on the other hand, embraces this quality correctly and has the right formulation for the issues regarding this quality.

Security: The SIV-method and the STACE framework have a lack/shortage of issues/criteria's regarding this quality. The IEEE RPSA on the other hand, embraces this quality correctly and has the right formulation for the issues regarding this quality.

Infological relations

Qualities **Arguments for integration of methods regarding qualities**

Intelligibility: This quality is covered by all three methods but the SIV-method and the IEEE RPSA embraces this quality well and has the right formulation for the issues regarding this quality.

Relevance: The IEEE RPSA and the STACE framework have a lack of issues/criteria's regarding this quality. The SIV-method on the other hand, embraces this quality correctly and has the right formulation for the issue regarding this quality.

Knowledge: The SIV-method and the IEEE RPSA framework have a lack of issues/criteria's regarding this quality. The STACE framework on the other hand, embraces this quality correctly and has the right formulation for the issue regarding this quality.

Competence: The SIV-method and the STACE framework doesn't pay attention to this quality. The IEEE RPSA on the other hand, embraces this quality but has a different slant on the formulation for the issue regarding this quality.

Motivation: This quality is embraced by the IEEE RPSA. The SIV-method doesn't pay attention to this quality and the STACE framework has a lack of issues/criteria's regarding this quality.

None of the methods fully fulfils all the relations, however all the methods fully fulfil the functional relations criteria's. IEEE RPSA is the only method that also fully fulfils the structural relations criteria's. The figure below shows how well the methods fulfils the functional, structural and infological need of knowledge.

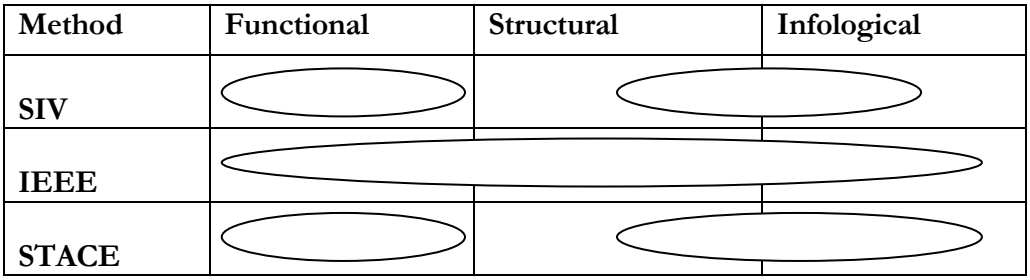

Figure 6.2 Available knowledge in chosen methods

6.3 The Meta-Architectural Product

This meta-architecture aims to function as support and guidance when in a situation of choosing standard system. Current approaches and methods for software evaluation do not adequately deal with these human, social, and organizational issues. Our solution to this is a meta-architecture that we have chosen to name MASSE, Meta-Architecture for Standard Systems Evaluation. The MASSE meta-architecture embraces the advantages of several existing software evaluation methods and particularly their infological, structural, functional, procedural and substantive issues, in one consistent meta-architecture. Unlike the other methods analyzed MASSE fully fulfills the functional, structural and infological aspects according to this thesis.

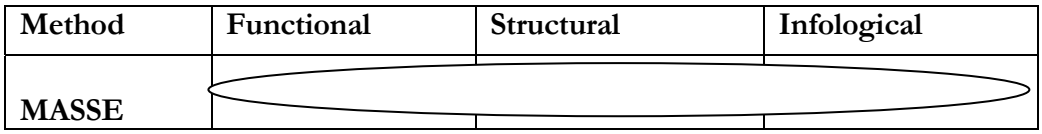

Figure 6.3 Available knowledge in the designed meta-architecture, MASSE

Several features separate MASSE from similar methods. The main feature is a suggested set of criteria's that expands the understanding and evaluation of characteristics beyond commonly evaluated characteristics such as function or cost. Categories of criteria include infological, functional and structural issues, which contributes to increased apprehension of the quality in the information system, and substantive and procedural issues, which contributes to increased security in the development work. Finally, the improved integration of knowledge addresses the issues of comprehensibility, understandability and meaningfulness.

6.4 Classification of MASSE

MASSE is a meta-architecture primarily developed from out of a combination of specific and general methods, which according to Nilsson's classifications complies with the description of a mutual method. The fact that MASSE also contains a comprehensive part as well as a more specific part where one can obtain deeper knowledge within in a certain area further contributes to us classifying MASSE within the field of a mutual method. Regarding method range we have chosen to put MASSE within the field of a minimal

method since we believe one cannot make any reductions to it and still obtain its full quality, however it is still possible to make add-ons to make MASSE more suitable to a specific situation.

Regarding how MASSE interacts with hardware and supplier issues we have found MASSE within the freestanding and separate areas of method-classification. Thus meaning MASSE doesn't pay any attention to neither hardware nor supplier issues. However MASSE takes notification in substantial issues but does not support the process of finding them. This means that the substantial issues could contain issues regarding both hardware and supplier.

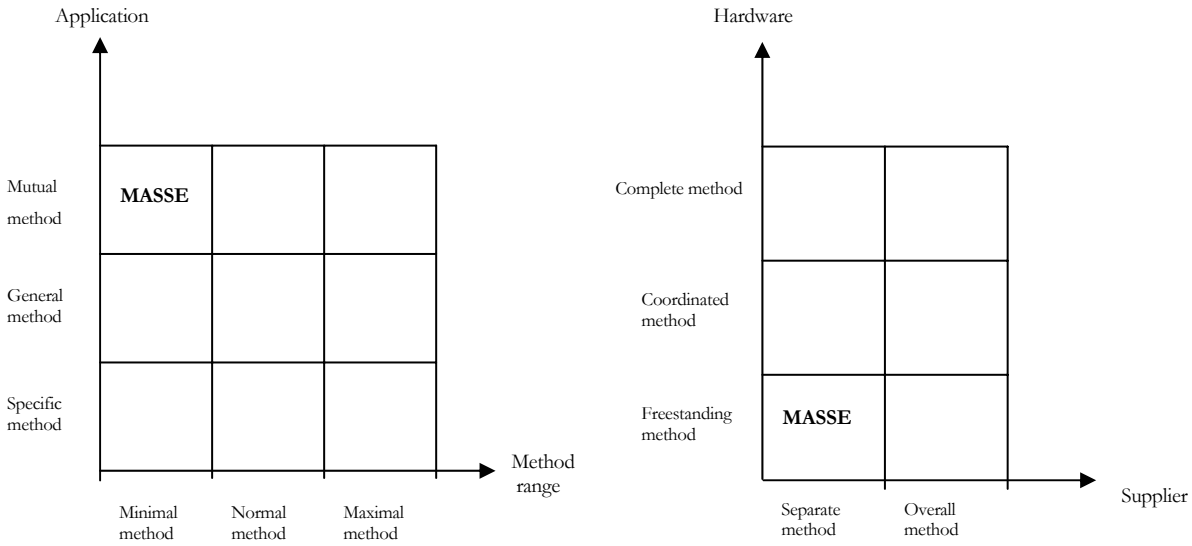

Figure 6.4 Classification of the designed meta-architecture, MASSE

7 Discussion

7.1 Problem

When we was assigned this master thesis we felt very insecure of having to make such an important choice as choosing a suitable standard system for OSS Sales. We had some knowledge in system development and information systems but none in evaluation methods and standard systems. We didn't know what evaluation method to use, what knowledge that was provided by the methods, what knowledge that was needed to make a sound choice, and what knowledge the used evaluation method should provide. We found that insecurity indicates the lack of knowledge within concerned areas. Our insecurity resulted as purposes and issues for further examination in this master thesis.

7.2 Method

In our thesis we investigate how to make a conscious and sound choice when acquisitioning a standard system. To empirically test our meta-architecture we applied it on a real case at the OSS Sales division at Ericsson, Gothenburg.

We consider our thesis to be of qualitative nature since a qualitative inception primarily is characterized by its closeness to the studied object. The reason for us calling it qualitative is quite complicated but originates from the fact that we functioned both as developers of the meta-architectural support and at the same time as subjects of empirical studies. One can say our meta-architecture, MASSE, is developed with the help of continuous and close consultations in shape of daily discussions with each other and also once in a while with our supervisor to reach states of mutual understanding. This is of course also a limitation in MASSE since MASSE being based on knowledge we personally have found indispensable. Another aspect we found to be important to be aware of while using professional conversations is to be really clear and discuss possible cases of misinterpretations. We believe there is more to win than to lose from that even if more time is consumed during the discussions.

In our thesis we have worked mostly inductively meaning that we have started out from a real problem situation from which we prior to this thesis made a present situation analysis and a requirement specification out of. Our inductive inception resulted in a meta-architectural support of how to create security in a situation of uncertainty of how to chose a standard system. We found the inductive inception of science to be the most feasible for us since we were working towards a theoretical point rather than from a theoretical point. However we tested our meta-architecture from out of the already made requirement specification and consequently we then also worked a bit deductive.

7.3 Results

Feeling insecurity made us understand within which areas we had a lack of knowledge. With the help of theoretical studies, scientific methods and analysis we gained needed knowledge. By analyzing the evaluation methods we noticed certain limitations within the methods. However, despite these limitations, both theories and methods can play a significant role in the awareness of the critical factors upon which a sound choice of information systems and information technology must be based. By combining method-fragments we could create a

more secure and confident meta-architecture, MASSE. One can say that we created security in a situation that we before felt insecurity within.

The empirical part of this thesis has two directions. The first direction is when we created MASSE. The creation of MASSE was made by us, as the designers, we also played another part during the development and that was to account for the reliability of MASSE. In that way we functioned both as the designers and subjects of empirical testing. This is clearly a limitation within the empirical study of MASSE as a meta-architecture that we are fully aware of. We therefore believe it would be advantageous if MASSE were tested more objectively, i.e. tested by a number of system developers. A proposal for future research would therefore be supplementing objective empirical studies of MASSE.

The other direction of empirical nature is when we tested MASSE on a real case at OSS Sales. OSS Sales wanted us to evaluate a number of request handling systems and propose a sound choice of a system. The OSS Sales situation only implied the creation of an underlay for decision of change. We therefore mostly focused on an evaluation among existing standard systems within Ericsson suitable for OSS Sales. Therefore, the procedural issues were limited to only "management of decision of change" and the evaluation focuses on the substantial issues.

However, besides the fact that the OSS Sales situation didn't need to evaluate procedural issues the evaluation was still not ideal. The requirement specification we used was made by us prior to this thesis for OSS Sales according to Ericsson standards. The requirement specification therefore didn't contain any issues of procedural nature. We also found while using the requirement specification for the standard system evaluation for OSS Sales that the requirement specification was made paying no attention to the needs and wants of the customers to this system. Therefore we believe there is a chance that the outcome of the evaluation could have been slightly different to what we found.

7.4 Conclusions

It is our belief that every developmental situation is unique. This uniqueness cannot be approached and fully understood by abstract theories and generic methods. Furthermore, theories and methods represent past experiences. Therefore their relevancy to the design and development of the future of a particular enterprise may be every limited. Many ready made rigid methods and so-called "universal" meta-architectures can lead to blindness because they can constrain creativity and innovativeness. However, despite these limitations, both theories and methods can play a significant role in the awareness of the critical factors upon which a sound choice of information systems and information technology must be based. Thus it is neither the true value of theories nor the efficiencies of methods that motivate their use in information system development. Instead their practical value is derived from the ability of managers, designers and developers to use them and generate attractive future worlds and/or provoke necessary changes in critical areas of enterprise and information systems.

7.5 Suggestions for future research

Would we have reached a different result if the made empirical study were of objective nature? A large-scaled empirical study of our meta-architecture, MASSE, is not within the frame of this master thesis and remains as an issue for future research.

8 References

8.1 Printed

Avison, D. E. and Fitzgerald G. (1995). *Information system development: methodologies, techniques and tools.* The McGraw – Hill Companies.

Backman, J. (1998). *Rapporter och uppsatser.* Studentlitteratur, Lund.

Bergenstjerna, M. (2002). *Towards the Design of a Meta-Architecture for IT-Management.* Department of Informatics, University of Gothenburg.

Bergenstjerna, M. et al. (1999). *Metoder för strategisk IT-management.* Master Thesis, Department of Informatics, University of Gothenburg.

Beynon-Davies, P. (1999). *Human error and information system failure: the case of the London ambulance service computer-aided despatch system project.* Interacting with Computers 11.

Brinkkemper S., Motoshi S. & Harmsen F. (1998). *Assembly Techniques for Method Engineering.* In Proceedings of the Tenth International Conference on Advanced Information Systems Engineering (CAiSE*98), Pernici B & Thanos C (ed.). Pisa, Italy.

Brinkkemper, S. (1996). *Method Engineering*: Engineering of Information Systems Development Methods and Tools. In Proceedings of the Fifth International Conference on Information Systems Development (ISD'96), Wrycza-Zupancic (ed.). Gdansk, Poland.

Buckingham, R. A. et al. (1987b). *Information systems curriculum: a basis for course design.* CUP, Cambridge.

Checkland, P. and Sholes, J. (1990). *Soft Systems Methodology in Action.* Chichester, UK, Wiley.

Checkland, P. (1981). *Systems Thinking, Systems Practice.* Chichester, UK, Wiley.

Churchman, C. W. (1979). *The Systems Approach and its Enemies.* New York, Basic Books, Inc Publishers.

Davidson, B., Patel, R. (1991). *Forskningsmetodikens grunder – Att planera, genomföra och rapportera en undersökning.* Lund, Studentlitteratur.

Dexner, P. (1995). *Administrativa standardsystem–en del av IT-strategin*. Studentlitteratur Lund

Ejvegård, R (1996). *Vetenskaplig metod.* Studentlitteratur Lund

Enquist H., Magoulas T., Bergenstjerna M. and Holmqvist M. (2001). *Delta Meta Architecture – Final Report.*

Enquist, H. and Makrygiannis, N. (1998). *Understanding misunderstandings.* 31st International Conference on Systems Science, Hawaii, IEEE Computer Society.

Eriksson, L. T. and Wiedersheim-Paul, F. (1999). *Att utreda forska och rapportera.* Malmö, Liber AB.

Friedman, B. and Kahn, Jr. P. H. (1994). *Educating Computer Scientists: Linking the social and technical.* Communication of the ACM, Vol. 37

Fåhraeus, E. (1986). In Swedish: *Metodkedjornas och verktygens roll vid systemutveckling.* Rapport nr 24 Riksdataförbundet, Stockholm.

Goldkuhl G, Lind M and Seigerroth U. (1997). *Method Integration as a Learning Process.* In Proceedings of the 5th BCS Conference on Training and Education of Methodology Practitioners and Researchers. Jayaratna N, Wood-Harper T & Fitzgerald B (eds.). Preston, England.

Goldkuhl, G. (1991). In Swedish: *Stöd och struktur i systemutvecklingsprocessen.* Institutionen för datavetenskap, Linköpings universitet.

Hage A., Kerim S. and Zamirian K. (2002). *Systemsamverkan – En studie av LADOK-systemets utformning och användning inom Göteborgs universitet och Centrala studiestödsnämnden.* Master Thesis, Department of Informatics, University of Gothenburg.

Halvorsen, K. (1992). *Samhällsvetenskaplig metod.* Lund, Studentlitteratur.

Harmsen, A. F. (1997). *Situational Method Engineering.* PhD dissertation. Moret Ernst & Young Management Consultants, Utrecht, The Nederlands.

Hedberg, B (1980). Using Computerized Information Systems to Design Better Organization and Jobs. *The Human Side of Information Processing.* N. Björn –Andersen, North-Holland.

Hoffman, T. (1988). Corporate Information Systems Strategy. *Information Systems in Practice and Theory.* P. Pirow, N. Duffy and J. Ford, Elsevier Science Publishers B. V. (North-Holland)

Hägerfors, A (1994). *Co-learning in participative systems design, Enhancements of genuine participation by consideration of communication and group dynamics.* Lunds University.

Högberg, K. (2002). *Kritiska framgångsfaktorer för IT inom vården: En studie ur vårdpersonalens perspektiv.* Magisteruppsats. Instutionen för informatik, Handelshögskolan vid Göteborgs universitet.

Jayaratna, N. (1994). *Understanding and Evaluating Methodologies.* McGraw-Hill Book Company, London.

Kunda, D. and Brooks, L.. (1999). *Applying social-technical approach for COTS selection.* Proceedings of 4th UKAIS Conference, University of York, McGraw Hill.

Lundahl, U. and Skärvad, P.H. (1999). *Utredningsmetodik för samhällsvetare och ekonomer*, Lund, Studentlitteratur.

Magoulas, T and Pessi, K. (1998). *Strategisk IT-Management.* Vasastadens bokbinderi, V Frölunda.

Mathiassen, L. (1982). In Danish: *Systemudvikling og systemudviklingsmetode, doktorsavhandling.* Datalogisk afdelning, matematisk institut, Aarhus universitet.

Nilsson, A. G. (1998). *The Business Developer's Toolbox: Chains and Alliances between Established Methods.* In Perspectives on Business Modelling - Understanding and Changing Organisations, Nilsson A G, Tolis C & Nellbron C (eds.). Springer Verlag, Heidelberg.

Nilsson, A. (1991). In Swedish: *Anskaffning av standardsystem för att utveckla verksamheter, doktorsavhandling*. Ekonomiska forskningsinstitutet (EFI), Handelshögskolan i Stockholm.

Nurminen, M. (1988). *People or Computers: Three Ways of looking at Information Systems*. Lund, Studentlitteratur.

Odum, H. T. (1996). *Environmental accounting.* John Wiley & Sons.

Potts, C. (1993). *Software engineering research revisited.* IEEE Software.

Rumbaugh, J. et al. (1991). *Object-Oriented Modeling and Design*. Englewood Cliffs, New Jersey: Prentice Hall.

Röstlinger, A. and Goldkuhl, G. (1994). In Swedish: *Generisk flexibilitet – På väg mot en komponentbaserad metodsyn*. Presenterat på VITS Höstseminarium 1994. Institutionen för datavetenskap, Linköpings universitet.

Sommerville, I. (1995). *Software Engineering.* Addison Wesley Longman Limited, Essex.

Stamper, R. (1988). *Analysing the Cultural Impact of a System.* International Journal of Information Management, vol. 8, no. 3.

Yin, R., 1994. *Case Study Research. Design and Method (2:nd edition).* SAGE Publications, London. ISBN 0-8039-5663-0

Yourdon, E. (1989). *Modern structured analysis*. Englewood Cliffs, New Jersey: Prentice Hall.

Öberg, F. (1998). *Object-Oriented Frameworks - A New Strategy for CASE Tool Development.*
Licentiate Thesis. Department of Computer and Information Science. Linköping University.

8.2 Unprinted

Arilla (1999). *Informationshantering.*
URL: http://www.arilla.se/informationshantering.htm // 2002-10-10

ASQ, (2002). *ASQ.*
URL: http://asq.ericsson.se //2002-08-29

EGIS, 2001. Ericsson Global IT Services pricelist year 2001 for global products
URL: http://itservices.ericsson.se/products/pricelist/word/pricelist_2001.doc // 2002-08-07

EGIS, (2001). *WIH.*
URL: http://itservices.ericsson.se/products/sheets/wih.html //2002-08-29

EGIS, 2002c. ESGP and mail size limitations.
URL: http://mail.ericsson.se/information/Limits/globalsizelimit.shtml //2002-08-07

Ericsson AB (2002b). *Ericsson in brief.*
URL: http://www.ericsson.com/about/compfacts/ // 2002-09-10

Ericsson. (2002). Info Sverige.
URL: http://infosverige.ericsson.se/it_stod/our_it_environment/mail/index_eng.asp // 2002-08-07

Ericsson University. (2002).
URL: http://university.ericsson.se // 2002-08-07

ESOE. (2002). *Software certification database*
URL: http://esoedb.ericsson.se/default.asp?type=SW //2002-08-07

Fallström, Johan. (2001a). *GS3 System Recovery: System Recovery Work Instruction.*
URL:
http://www.lmc.ericsson.se/gec_mtl/operations/process/documents/lmc_ytw/Tools_Sup_Adm/GS3_system_recovery_WI.doc // 2002-08-08

Fallström, Johan. (2001b). *GS3 O&M Work Instruction.*
URL:
http://www.lmc.ericsson.se/gec_mtl/operations/process/documents/lmc_ytw/Tools_Sup_Adm/GS3_OM_WI.doc // 2002-08-09

Global Services. (2002). *Service Management Services Project: Frequently Asked Questions.*
URL: http://globalservices.ericsson.se/cs/e-supportsystems/faq/index.htm // 2002-08-08

GS3 Support Team. (2002a). *GS3, Global Services & Support System.*
URL: http://gs3.ericsson.se/ // 2002-08-08

GS3 Support Team. (2002b). *GS3 Product Information.*
URL: http://gs3.ericsson.se/customer_docs/prodinfo.shtm // 2002-08-08

Halderman, R. (1998). *GS3 presentation, Rev B.*
URL: http://gs3.ericsson.se/customer_docs/gs3_doc/GS3revb.ppt // 2002-08-08

Joslyn, C. and Turchin, V. (1993). *System.*
URL: http://pespmc1.vub.ac.be/^system.html // 2002-10-10

Joste, Ginta. (2002). *Progress Report for Service Management System (SMS) Project, 31st – 31st July 2002.*
URL: http://globalservices.ericsson.se/cs/e-supportsystems/progress/sms/VG-02-0192-A%20Progress%20Report%20SMS%20Project%20July%202002.doc // 2002-08-16

Kallerman, Patric. (1999). *GS3 User Interface Requirement Specification.*
URL: http://gs3.ericsson.se/customer_docs/gs3_doc/Gs3rsa95.doc // 2002-08-09

LM Ericsson. (2002). *SMS Training.*
URL: http://university.ericsson.se/share/sms/ // 2002-08-12

Lowegard, Peter. (2002). *Service Management Systems (SMS): PC HW & SW requirements.*
URL: http://globalservices.ericsson.se/cs/e-supportsystems/faq/SMS_PC_requirements_B.doc // 2002-08-16

Method & Tools, (2002). *Method and Tools.*
URL: http://mt.ericsson.se //2002-08-29

Magnusson, Jens. (2000). *Final Repport for: GS3 R3A.*
URL: http://bokhyllan.ericsson.se/organisation/i/Product_development/utveckling/final_report_GS3_R3A.pdf
// 2002-08-09

Nordenborg, Lennart. (1996). *The GPMS project: A short introduction.*
URL: http://webmasters.exu.ericsson.se/projects/GPMS/cs96175.doc // 2002-08-08

SAP. (2000). *mySAP.com Workplace: Configuration and Administration Overview.*
URL:
http://epac.ericsson.se/solutions/hrms/download/sapdocs/Configuration_and_Administration_Overview.pdf //
2002-08-12

SMS Deployment. (2002). *SMS User Guide.*
URL: http://globalservices.ericsson.se/cs/e-supportsystems/smsproject/deployment/SMS_UserGuide-WIPv1.1.doc // 2002-08-14

SMS Functional Team. (2002). *SMS Detailed Functional Specification - Updated Requirements.*
URL: http://globalservices.ericsson.se/cs/e-supportsystems/smsproject/docs/__VG-02-0166-B_SMS_Detailed Functional Specification Updated Requirements.doc // 2002-08-13

Stolpe, Torbjörn. (2000). *GS3 – CSR Handling.*
URL:
http://insideuk.ericsson.se/solutiondelivery/mso/gate/documents/03812_GS3_CSR_Handling_Manual_PB3.pdf // 2002-08-19

Stolpe, Torbjörn. (1999). *GS3 – Manual.*
URL: http://gs3.ericsson.se/customer_docs/gs3_doc/GS3_manual_PA2.doc // 2002-08-09

Wessbrandt, K. (2002). *Förstudierapport: IT-plattformar för informationsförsörjning.*
URL: http://www.statskontoret.se/pdf/2002102.pdf // 2002-10-11

Willems, F., Pedersen, T. and Åkerlind, A. (2002). *Functional Verification Blueprint Gap Analysis.*
URL: http://globalservices.ericsson.se/cs/e-supportsystems/smsproject/docs/SMS%20Business%20Blueprint.doc
// 2002-08-12

Zackrisson, Ch. (2001). *GS3 For New Users.*

URL: http://plmoss.lmera.ericsson.se/gs3/GS3_for_New_Users_Design.ppt // 2002-08-08

8.3 Personal contacts

Almroth, Kjell-Åke. *Product Manager: GS3*. ERAALMR. 2002-07-31.

Cederstrand, Bjarne. Manager Customer Relation Team EAB. Global IT services. 2002-06-04

Dilks, Michael. *System & Application Support: GS3*. EEAMED. 2002-07-05.

Hellberg, Morgan. *Design Manager*, Global Support Tools. 2002-07-22

Hexeberg-Schoultz, Ann-Charlotte. *Dispatcher*. EMWAHEX. 2002-07-04.

in't Groen, Kristel. *Base Data Admin: GS3*. ETMKRIG. 2002-07-31.

Olsson, Gunilla. *ASQ Agent*. 2002-07-12

Rönnberg, Agneta. *Global IT Services*.GIS. 2002-08-05

Appendix

Appendix A – Requirement Specification

Functional Requirement Matrix

Requirement	Slogan	Priority
Input		
R1	Ability to send requests via web-portal	1
R2	Ability to send requests via e-mail	2
R3	Ability for the requester to attach files to the request	1
Usability		
R4	Improvement in functionality and usability	1
R5	Automatically filled out fields (requester information)	2
R6	Option to view receipts before they are sent	3
R7	Possibility to work towards a web-GUI	1
R8	Ability for the requester to check the request-status on the web	2
R9	Option for the handler to show his/her name	3
R10	Possibility to modify GUI and functionality	1
R11	Ability to log activities	1
R12	Ability to show statistics: see ref.	1
R13	Possibility to view handler status	1
R14	Possibility to view dispatcher schedule	3
R15	Ability for the dispatcher to reclaim the rights of a task	1
R16	Possibility to write a comment on decline of task	2
R17	Preview list on tasks	2
R18	Ability to write working log	2
R19	Predefined names on experts to help handler	2
R20	Easy to find, sort and view old cases	1
System		
R21	Automatically generated caseID	1
R22	Read-Only caseID	1
R23	Integration with LDAP or similar	2
R24	Automatically updated cases if new information is acquired	1
R25	Alarm if case appears to be forgotten	1
R26	Database indexed by: see ref.	1
R27	Not be able to close a case without commenting	3
R28	Incoming e-mail support	1
R29	Outgoing e-mail support	1
R30	Possibility to use the system offline	2
R31	Shared knowledge-database with other support-units	3
R32	Public holiday control	2
Output		
R33	Automatically generated receipts	1
R34	Automatically generated answers from templates	1
R35	Ability for the handler to attach files to the solution	1

Priority 1: Means that it is a very important criteria that is primary
Priority 2: These criteria's facilitate the work if they are fulfilled
Priority 3: It is not a necessity, but is in some cases useful

Functional Requirements Analysis

Function R1: Ability to send requests via web-portal

Description

There shall be a web-interface so that the requester can fill in a request via a web-portal. The input form should consist of certain fields required by OSS Sales.

Scope

Using a web-portal to submit a request the customer can be forced to fill in certain fields that are of importance to OSS Sales. This way of procedure minimizes the manual administrative work of the handler because the incomming request is automatically registrered into the Request Handling System.

Function R2: Ability to send requests via e-mail

Description

The requester shall be able to send requests via e-mail.

Scope

Since most requesters are used to send requests via e-mail there will probably be incomming e-mail requests under a period of time. These requests must not be ignored and therefore there shall be a possibility for the requsters to send requests via e-mail.

Function R3: Ability for the requester to attach files to the request

Description

The requester shall be able to attach files to the request in case important information needs to be enclosed with the request.

Scope

In some situations the requester needs to enclose binary information to the request (i. e. pdf documents).

Function R4: Improvement in functionality and usability

Description

Improvement in functionality and usability of todays Request Handling System.

Scope

The new system must give prominence in functionality and usability compared to the old system.

Function R5: Automatically filled out fields (requester information)

Description

The requester shall only have to fill in his corporateID and the rest of the fields regarding personal information are automatically filled out.

Scope

This way of procedure minimizes the manual administrative work of both the requester and handler.

Function R6: Option to view receipts before they are sent

Description

There shall be an option for the dispatcher to view and edit the automatically generated receipt before it is sent.

Scope

Some requesters needs a more personalized receipt. Therefore there shall be a possibility for the dispatcher to edit the automatically generated receipt before it is sent.

Function R7: Possibility to work towards a web-GUI
Description
There shall be a possibility for the handler/dispatcher to work towards a web-GUI.
Scope
Web-GUI is a popular and userfriendly interface.

Function R8: Ability for the requester to check the request-status on the web.
Description
There shall be an easy way for the requester to view the status of a specific request on the web.
Scope
To avoid extra work and waste of time concerning status of a specific request the requester shall have the ability to check the status of the request on the web.

Function R9: Option for the handler to show his/her name
Description
There shall be a function that allows the handler to either show his/her name or to be anonymous in the solution.
Scope
When showing your name as a handler to a task there seem to be a big risk that the requester will send future requests directly to a you.

Function R10: Possibility to modify GUI and functionality
Description
There shall be a possibility to modify the GUI or functionality in the system if there is a need for it. This is for the system administrator only.
Scope
The future demands for GUI and functionality may change. There must be ways to comply with this. Therefore there shall be a possibility to modify the GUI or functionality in the system.

Function R11: Ability to log activities
Description
The system shall be able to log activities that may be of importance for statistics.
Scope
It is of importance to log activities for statistics and tracebility.

Function R12: Ability to show statistics
Description
There shall be a way for concerned users to get statistics out of the system. The desired statistics areas are:

- Regional differences
- What kind of questions
- Which accounts
- Products
- Optional features
- Areas of technology

When a task is initiated and closed. How many working hours that are put into different tasks.
Scope
It is of imporatance to get statisctics out of the system to be able to show results and indentify trends.

Function R13: Possibility to view handler status
Description
The dispatcher shall be able to view how many cases and witch cases a certain handler is working with at the moment.
Scope
This function facilitates the dispatchers work in finding a suitable handler.

Function R14: Possibility to view dispatcher schedule
Description
There shall be a possibility to view a dispatcher schedule so that the handlers can see future/history dispatcher.
Scope
Easily accessible overview of a dispatcher schedule is needed to know when a certain person was or is going to be dispatcher.

Function R15: Ability for the dispatcher to reclaim the rights of a task
Description
There shall be a possibility for the dispatcher to reclaim the rights of a task so that no task is left pending because the assigned handler is out of office.
Scope
As a dispatcher you must have the possibility to take back the rights of a task. This is useful when to avoid a task to be left pending and letting precious time go to waste if the assigned person is out of office and is not able to work with the task.

Function R16: Possibility to write a comment on decline of task
Description
If a task is to be declined the declining handler shall have the possibility to comment the reason for the decline.
Scope
This function enables the handler to motivate why he/she is declining an assignment.

Function R17: Preview list on tasks
Description
There shall be a function that allows handlers/dispatchers to view a list of active/closed cases with a little preview of each.
Scope
A good overview of active/closed task facilitates the work of a dispatcher/handler.

Function R18: Ability to write working log
Description
The handler shall be able to write a working log while working on a case.
Scope
This function helps the handler to keep track of what has been done and what is to be done in a case.

Function R19: Predefined names on experts to help handler
Description
Predefined names on technical experts. So one can easily know who to contact for support.
Scope
A list of experts connected to expertareas facilitates the handlers work in trying to find expertise knowledge.

Function R20: Easy to find and sort and old cases
Description
It shall be posslible to store requests in a database. There shall be a good solution for the database so one can easily and fast find old cases.
Scope
Facilitates the work when trying to find old cases.

Function R21: Automatically generated caseID
Description
Automatic numbering of cases, each case gets its own unique number.
Scope
This helps avoiding problems regarding typing errors which makes future searching more difficult. This function also facilitates the manual work of the dispatcher.

Function R22: Read-Only caseID
Description
The caseID shall be Read-Only. Because each task gets its own uniqe number this ID-tag shall not be editable.
Scope
This helps avoiding problems regarding typing errors which makes future searching more difficult

Function R23: Integration with LDAP or similar
Description
The input-form on the web portal, and the Request Handling System shall be integrated with LDAP or similar database.
Scope
A connection with a personal database facilitates the administrative work by automatically filling in personal information about the requester.

Function R24: Automatically updated cases if new information is acquired
Description
If new information is acquired about a specific case, this case shall automatically be updated and the handler for the case shall be notified about this.
Scope
This functions facilitates the handlers work if new information about an already existing case is received. Instead of inserting the new information manually to the case this is done automatically.

Function R25: Alarm if a task appears to be forgotten
Description
If a task is getting near its due-date, the handler shall be alarmed about the situation.
Scope
This function helps the handler not to forget a task.

Function R26: Database index
Description
The Database shall be indexed by:

- Customer
- Date (period)
- System
- Technology Area

- Country
- Continent
- Product
- Version
- OSS Package
- Optional Features
- LPM

Scope
This function enables the handler to make both free text search and combination search. Combination search exists of two or more combined search areas.

R27: Not be able to close a task without commenting
Description
As a handler not be able to close a task without commenting why closure is done.
Scope
It is of the dispatcher's interest to know why a task is closed.

R28: Incoming e-mail support
Description
The Request Handling System shall have support for incoming e-mails.
Scope
A part of the requests will come in to OSS Sales via e-mail and therefore there must be support for incoming e-mails.

R29: Outgoing e-mail support
Description
The Request Handling System shall have support for outgoing e-mails.
Scope
All solutions will be sent via e-mail and therefore there must be support for outgoing e-mails.

R30: Possibility to use the system offline
Description
Possibility to handle the system offline if needed.
S*cope*
This function is useful when traveling or in an other way working out of office.

R31: Shared knowledge-database with other support-units
Description
Sharing knowledge-database with other support-units
Scope
Sharing knowledge with other support-units minimizes the amout of work for solution to requests and minimizes the amout of double-work done by different support-units handling the same request from same requester.

R32: Public holiday control
Description
Automatic due-date modification when public holiday occurs so misunderstanding and frustration won't arise.
Scope

This function helps avoiding misunderstanding and frustration due to by the requester unexpected national public holidays.

R33: Automatically generated receipts
Description
Acknowledgement to customer that request have been received shall be automatically generated.
Scope
This function faciltates the administrative work of the dispatcher.

R34: Automatically generated answers from templates
Description
Templates for solutions shall be automatically generated with ability to edit before sending.
Scope
This function faciltates the administrative work of the handler.

R35: Ability for the handler to attach files to the solution
Description
The handler shall be able to attach files to the solution in case important information needs to be enclosed.
Scope
In some situations the handler needs to enclose binary information to the solution.

Technical Requirements Matrix

Requirement	**Slogan**	**Priority**
R36	System performance	1
R37	Software support	1
R38	System and hardware support	1
R39	Client platform requirement	1
R40	Server platform requirement	1
R41	Language requirement	1
R42	ESOE certification	1

Priority 1: Means that it is a very important criteria that is primary
Priority 2: These criteria's facilitate the work if they are fulfilled
Priority 3: It is not a necessity, but is in some cases useful

Technical Requirements Analysis

R36: System performance

Description
The system shall show an acceptable administrative performance.
Scope
The system must be quick and support multiple users without impact on system performance.

R37: Software support
Description
There shall be a reliable and accessible software support .
Scope
To fix and prevent software errors there must be a support unit for those kinds of issues.

R38: System and hardware support
Description
There shall be a reliable and accessible system and hardware support .
Scope
To fix and prevent system failures there must be a support unit for those kinds of issues.

R39: Client platform requirement
Description
The client software shall be supported on Windows (NT and 2000).
Scope
Most users are using MS Windows enviroment (NT and 2000) and therefore the client software must support this.

R40: Server platform requirement
Description
The server software shall be supported on Solaris (7 and 8) and MS Windows NT version 4.
Scope
Solaris (7 and 8) and MS Windows NT version 4 are supported by the IS/IT department, therefore the server software must support this.

R41: Language requirement
Description
English shall be supported in the RHS system.
Scope
English is the general language in the Ericsson concern.

R42: ESOE certification
Description
The system shall be ESOE certified.
Scope
Software within Ericsson should be cleared in ESOE (Ericsson Standard Office Environment) in order to be maintained properly by system administration. Approving a new software in ESOE takes a lot of time and should be avoided.

Appendix B – Present situation analysis (OSS Sales)

Operational Sales Support background information

OSS Sales Support covers Commercial Sales support and Technical Sales support. The purpose is to proactively plan and execute market activities, commercial and technical sales support of the product portfolio including all optional functions and services. OSS Sales Managers (commercial responsibility) and Solution Managers (technical responsibility) maintain this task together, equally important. The purpose is to optimize and secure market share in terms of new systems & upgrades, optional products and services, by controlling the sales process and to drive sales activities. How to focus and what functionality to promote is based on business decisions and customer needs. The OSS Sales Helpdesk has been created to make a single interface to handle OSS requests that are related to a business opportunity. The new organisation put new responsibilities to the OSS Sales Helpdesk. The product areas to be handled are: GSM-OSS, CN-OSS and Ranos. The OSS Sales Managers and Solution Managers are using proven sales methods like "Account Tracker"(Holden based). The OSS Sales Manager has a commercial responsibility, related to a specific number of important GSM customers or a defined BMOS Business & Sales Management unit. The Solution Manager has a product technical responsibility to be carried out both to identified customers and BMOS Business and Sales Management units. The solution Manager will hold a very important responsibility to implement a well-defined contact network between relevant PU organizations as CN and RANOS. In addition to this the Solution Manager will have a responsibility to take part of the development of the 3G OSS "story"/launch package and to carry out this to the market together with the OSS Sales Managers. In some cases OSS Sales Managers and or Solution Mangers will operate locally which means they will have a responsibility based on local prerequisites.

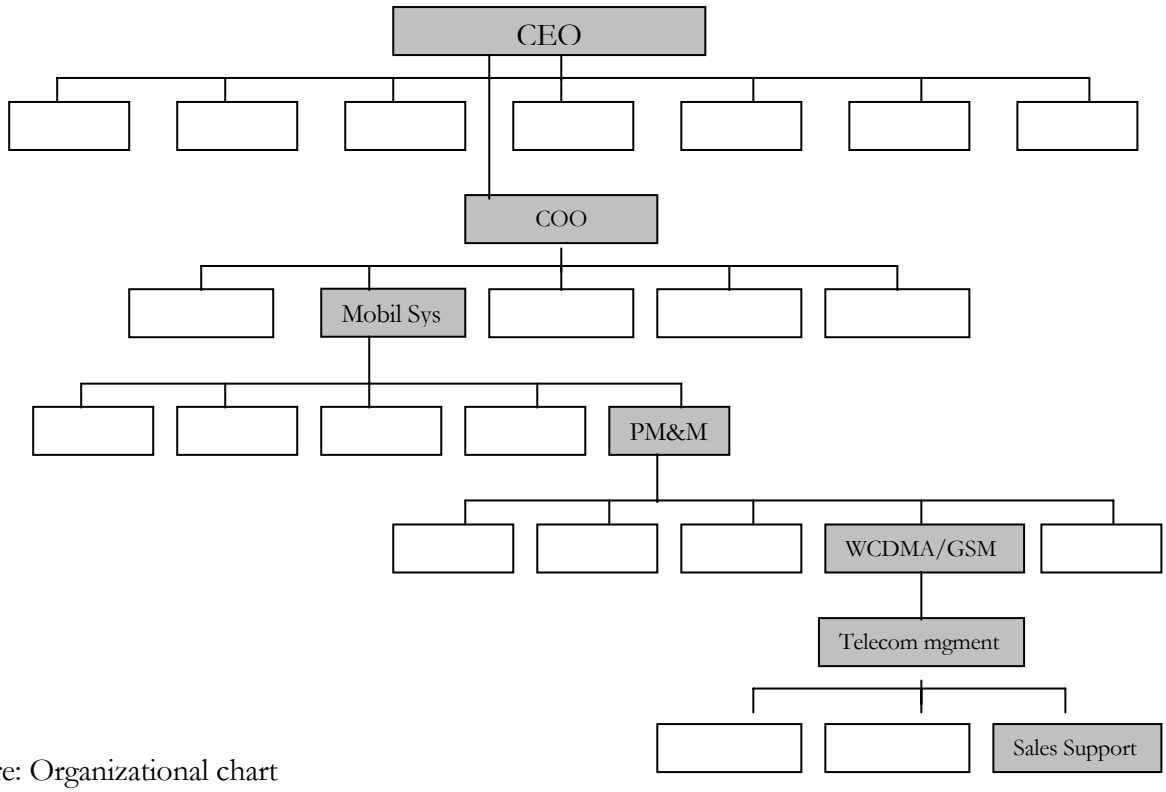

Figure: Organizational chart

Support handling process

Market Units and Key Account Managers

Ericsson has local market units, MU, represented in almost every country in the world. These MU have multiple business accounts. A business account is a connection to an Ericsson customer of larger scale. Behind every account there is an account team to promote the relationship between Ericsson and it's customers. The Business Manager (BM) / Account Manager (AM) at the MU is always responsible for the customer and the business. When expertise help is needed, the key-account manager within the MU makes a support request to OSS Sales (Oss.Sales@ericsson.se).

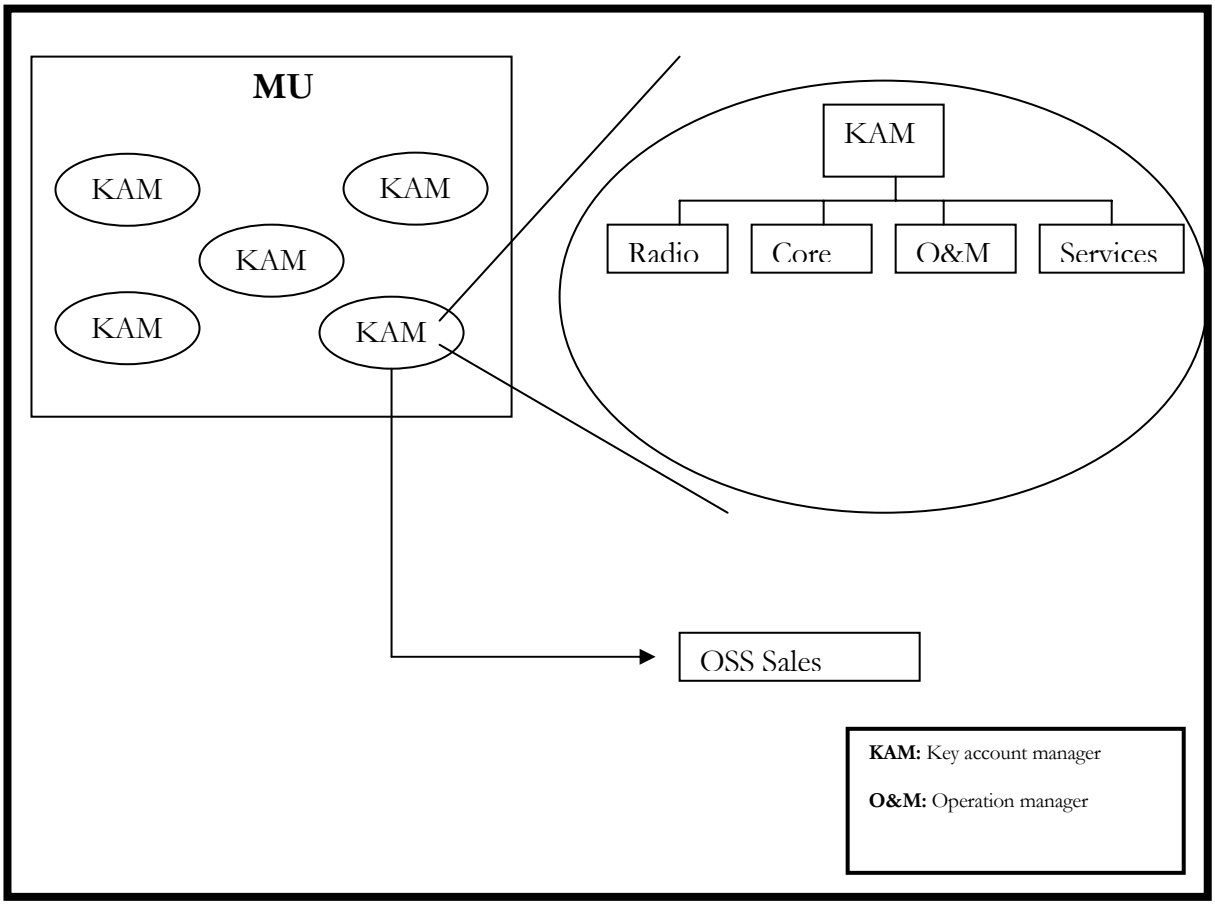

Figure: KAM within MU

Handlers

When receiving and accepting a request for support the task can be handed out to different handlers. The choice of which handler should be assigned the case depends on personal competence and time availability. When the task is finished the solution is handed over to the original customer (MU). A copy of the solution is also sent to OSS to function as a knowledge databank in case of similar requests in the future.

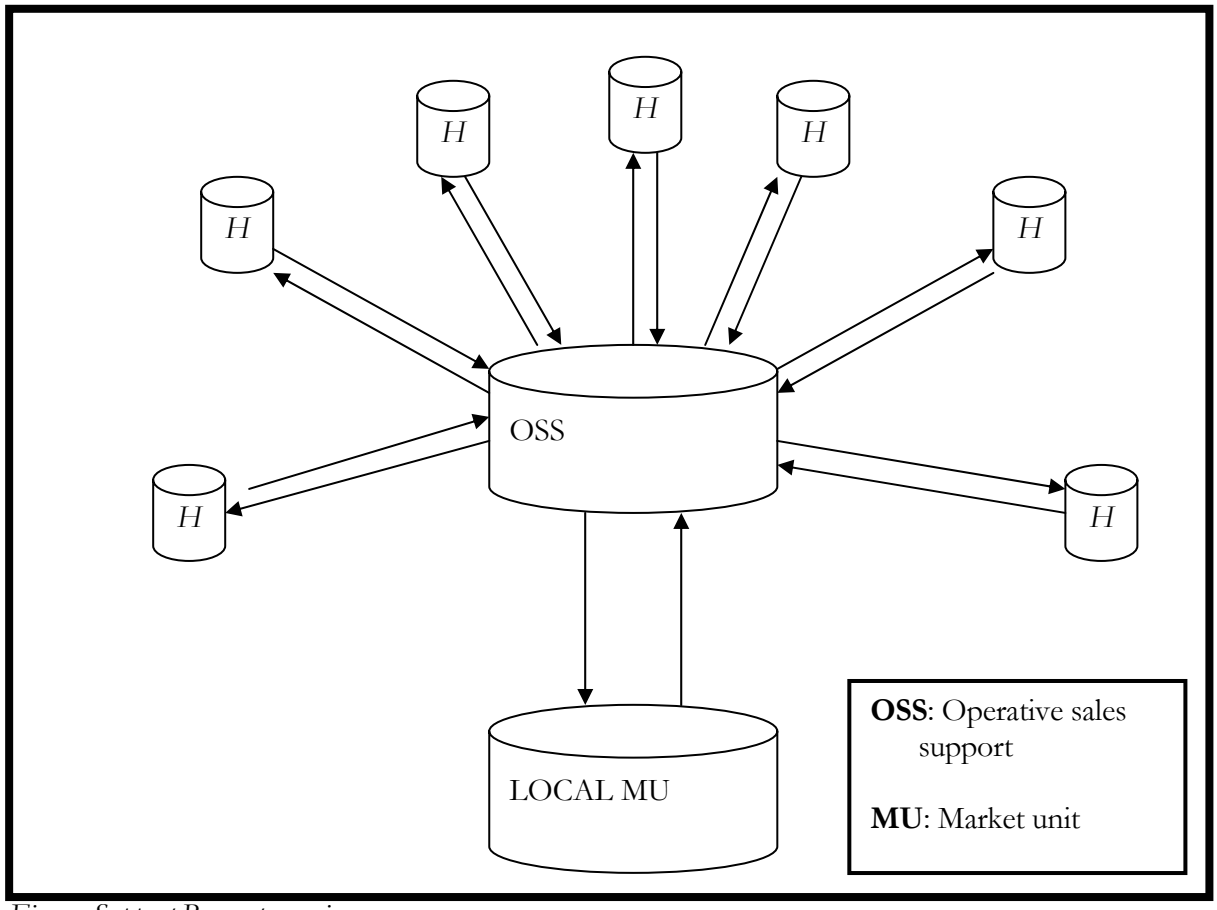

Figure: Support Request overview

Support Request Process

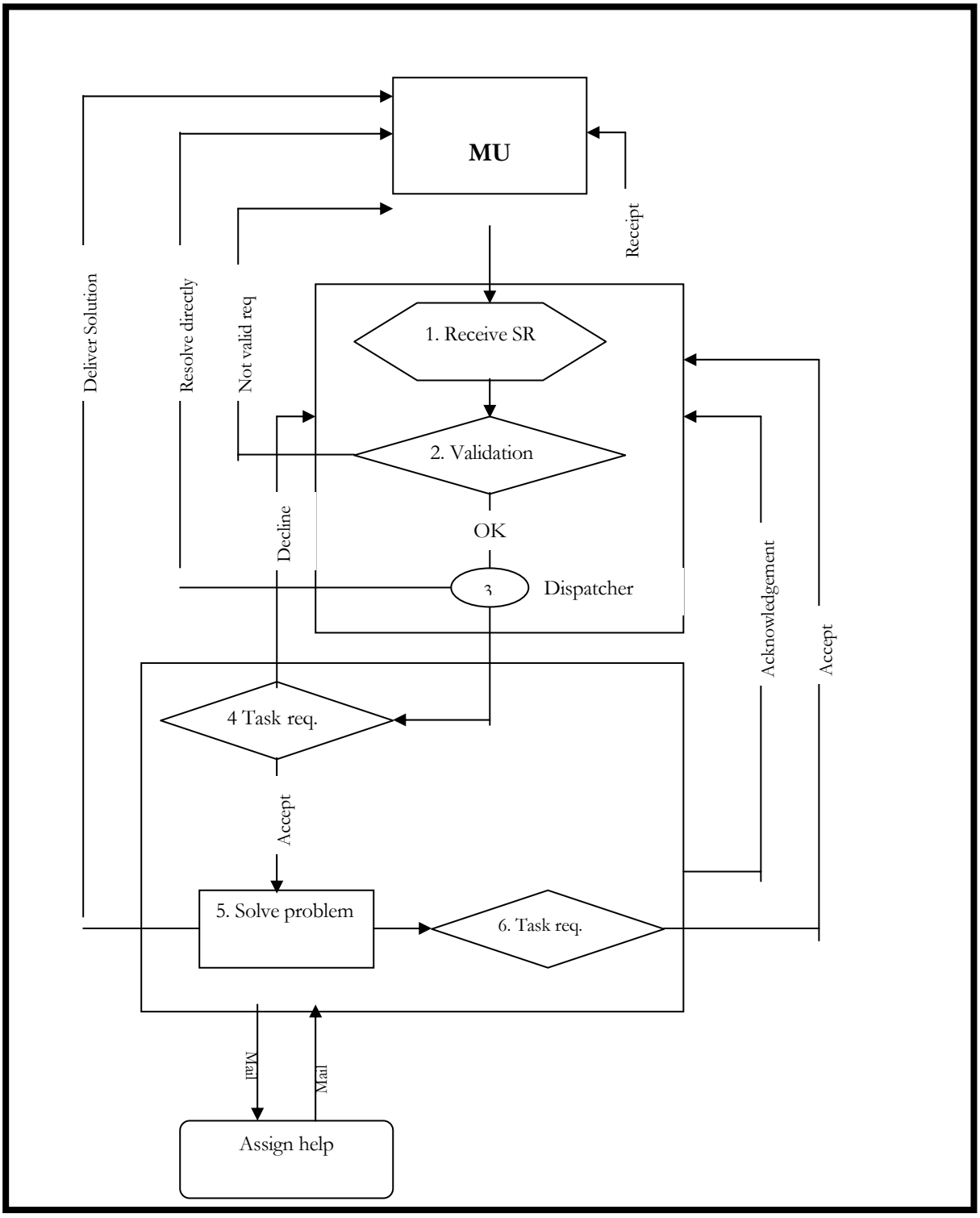

Figure: Support Request Process

Nr	Description	Input	Output
1	Reception of support request through OSS Sales e-mail address	Support request	Request for validation
2	The dispatcher makes an analysis of the request. The dispatcher must check if the request is valid within OSS product areas.	Support request	Rejection if the request is without any relation to a business opportunity or if the request is not within the correct management related working field. Rejection can also occur if there is a short deadline and the OSS doesn't have enough resources. The mail is returned with a rejection statement and directed to the correct helpdesk. Acceptance of the request with decision of Service Level related to RHS strategy. Dispatching to other product areas e.g. SOG, Billing. A receipt back to the origin of the request is sent.
3	The dispatcher ensures that he or she has the right competence and enough time to solve the problem or service request. If competence is lacking or severe time constraints exist, the problem or service request is assigned to another of OSS employees.	Support request	If the dispatcher can resolve the request directly, then solve problem or service request and make a direct answer to MU. Otherwise, assign the task to another of OSS employees that has the time and competence. Create a Task Request in the OSS Sales mailbox. Explanatory information about the task is inserted into the task. A reference ID is made.
4	If competence is lacking or severe time constraints exist, the OSS-employee can decline the task request. Otherwise, accept the task request and become a handler.	Support request Task request	Initiation of a new task request. Handler of a task, i.e. handler of a support-case.

5	This is the problem-solving phase. If the problem solver, handler, can't solve the problem by himself he can take help from outside expertise. This communication is managed by mail. After resolving problem or service request, deliver the correct solution or service to the customer (MU). The delivery is carried out manually through mail or telephone.	Knowledge database Old requests-cases Ongoing request Internal expertise	Documented solution to the support request. Solved problem or service request. Delivered solution or service to the customer (MU).
6	Assign task to dispatcher i.e. send a task request to dispatcher to give back the read/write rights of the task to OSS. Dispatcher accepts the task request and closes the task.	Solution of problem or service request	Documented solution to the support request.

Appendix C – Terms and Abbreviations

Definitions / ACRONYMS

The ESGP Concept
ESGP is the conceptual name of Ericsson Global IT Services services delivering mail and groupware functions to Ericsson users. There are two main components in the ESGP concept; MS Exchange and Outlook.MS Exchange is the program software installed on the server and Outlook is the client software used by the end user. All functions within ESGP may be used for personal efficiency or as a powerful tool within or between groups of individuals, independently of time zones, organisation or geographic situation. The functionalities included in ESGP are amongst others: e-mail, calendar, tasks, competence networks (news groups), pre formatted forms via e-mail, advanced possibilities to develop business supporting applications and other kinds of groupware. More information about ESGP can be found at: http://mail.ericsson.se/information/esgp/

GPMS - Global Problem Management System
GPMS origins from the GPMS project in the year 1996 which goal was to define a common Global Problem Management System platform. This platform was developed from Clarify ClearSupport and now functions as a development platform for Ericsson support organizations. From this platform each new organization joining can customize their own problem management system and thus become a part of a global Ericsson support structure. The benefits of this are primary cost issues of not having to build expensive individual local support systems and also the ability to share knowledge between participating organizations. More information about GPMS can be found at: http://webmasters.exu.ericsson.se/projects/GPMS/

EGIS - Ericsson Global IT Services
EGIS is the manager of IT services for Ericsson. EGIS provides cost effective ready to use standardized solutions with highly consolidated concepts covering all aspects from local site to central implementation. More information about EGIS can be found at: http://egis.ericsson.se

ESOE - Ericsson Standard Office Environment
ESOE is a collection of tools and components for the PC environment and is intended to function as a company-wide office environment for all PC-users within Ericsson. When installing only ESOE certified products you can be sure of not having any problems regarding hardware and/or software integration problems. This also helps reducing support costs.

With the ESOE client software in place, desktop and laptop PC users can store, retrieve and exchange data with other users worldwide using the same Network Operating System, as long as they are part of the ESOE program. As well as this, simply by logging in from an ESOE client PC, a user has access to their own customized working environment and their personal data files – from anywhere in the world. ESOE also supports upgrade and maintenance of core products from a central point. More information about ESOE can be found at
http://esoe.ericsson.se

Activities log
A summary of the case/subcase or solution, where each logged entry and action (e.g.: attachment added, change of status, change of owner) is recorded as one line.

Case
Basically the Electronic record within a Request Handling System of a CSR. Sometimes however, one CSR can result in more than one Case.

Case History
The main text-part that contains the stored log… entries in chronological order.

CSR - Customer Service Request
A question, complaint, Spare board request, Trouble report or similar from the customer.

ESP – Ericsson Service Portal
An electronic interface that connects GS3 to other GPMS databases, MSS databases and TRtool. It is used to send and receive CSR electronically with a minimum of re-typing.

ASO-AU
An ASO is a 2nd line support organisation within the BR organisation. ASO-AU is located in Australia and serves the Asia-Pacific markets for GSM.

BDA
Base Data Administrator. Responsible for the Base data handling, such as UserIDs, Queues, work groups, site and contracts etc.

GRC
Global Response Center. Public networks backbone support organisation and 2nd level support organisation. Geographically spread over: The Americas (Dallas), Asia/Pacific (Melbourne) and Africa & Europe (Rijen, Holland). GRC is the current owner of the infrastructure that will be used by the GS3 project.

GCSO
Global Customer Support Office. A CSO for Public networks global customers. Geographically co-located with GRC plus one extra office in United Kingdom.

ERA/Z
Radio OSS support and design. This will at least include the ASO, 2nd line support and design.

ECSI
Ericsson Communication Standard Interface. Interface to enable sending of cases between GPMS, MSS, TR-Tool and OUP.

ETX/A
A public networks design department for network management products (SMAS).

ETX/XT
Design department for Network solutions Internet products.

LDA
Local Data Administrator. Serves a similar purpose as the BDA but belongs to one workgroup only.

MHS
Message handling system. Ericsson global IBM/VMS based tool for product support (Trouble reports, correction handling etc).

MSS
CSR & package handling and interface towards MHS.

OUP
Open Up time. Customer support tool, including CSR handling. Used by some Ericsson organisations.

TR-Tool
Unix based interface to MHS.

RACOM
RACOM (Ericsson Corporate Global Remote Access) is a flexible product for dial-up communication to ECN (Ericsson Corporate Network). The service enables Ericsson employees to work while they are out of office. More information about RACOM can be found at http://racom.ericsson.se.

GS3 - Global Service & Support System
A customer service management system, designed to link Ericsson technical support organisations into a global support network. More information about GS3 can be found at: http://gs3.ericsson.se.

CSR Writer - Customer Support Request Writer
A web based application used in order to simplify and structure the communication between Ericsson first line support (ELS) and their customers. It offers the customers functionality to issue and follow up CSRs.

SCS-PRIMUS e-Support - Solution Centered Support Knowledge Base
Primus eServerWeb client will allow you to search, modify or create solutions in the knowledge base. eServerWeb is typically the tool used by engineers to search and update the solution database.

SMS - Service Management Systems
A system that will deliver a new 'global' CSR case handling process to the 1st and 2nd level support organisations within targeted Market Units. It is a module in SAP and will replace GS3 and MSS. It is not implemented yet. More information about SMS can be found at:
http://globalservices.ericsson.se/cs/e-supportsystems/smsproject/.

Abbreviations

CSR	Customer Service Request
ELS	Ericsson Local Support (1st line)
EGS	Ericsson Global Support (2nd line)
NS	Network Support
OSS	Operation Support System
GSM	Global System for Mobile Communication (Groupe Special Mobile)
GUI	Graphical User Interface
CN	Core Network
BMOS	Business Unit Mobile Systems
PU	Product Unit
3G	Third generation
LDAP	Lightweight Directory Access Protocol
TM	Telecom Management
RHS	Request Handling System
LPM	Local Product Management